



PROYECTO FIN DE CARRERA PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

TEMA: Gestión de proyectos

TÍTULO: Evaluación, instalación y configuración de herramientas de gestión de proyectos de desarrollo de software

AUTOR: Julio César Ripoll Moreno

TUTOR: Javier Malagón Hernández

Vº Bº.

DEPARTAMENTO: DIATEL

Miembros del Tribunal Calificador:

PRESIDENTE: Eduardo Nogueira Díaz

VOCAL: Javier Malagón Hernández

VOCAL SECRETARIO: Pablo Ramíres Ledesma

DIRECTOR:

Fecha de lectura: 24 de septiembre de 2012

Calificación:

El Secretario,

RESUMEN DEL PROYECTO:

Actualmente existen multitud de aplicaciones creadas para la gestión de proyectos software; cada una de ellas pretende dar solución y facilitar las tareas propias de los gestores y los desarrolladores pertenecientes a los equipos de desarrollo.

Los equipos de desarrollo software suelen estar integrados por gran variedad de recursos, tanto humanos como materiales. Cada uno desempeña una función concreta en el proyecto, pudiendo no tener una dedicación plena al proyecto. Por eso, es necesario que dichos recursos sean compartidos entre la cartera proyectos existentes. Para resolver este planteamiento en las aplicaciones de gestión de proyectos, ha sido requisito fundamental que se puedan gestionar varios proyectos de forma simultánea (gestión multiproyecto), pudiendo repartir la dedicación de los recursos entre los proyectos existentes en la cartera. En la actualidad, existe un gran número de metodologías de gestión de proyectos, por lo que, en parte, el éxito del proyecto radica en la elección de la más adecuada. Entre todas las metodologías existentes, este estudio se ha centrado en las cada vez más utilizadas metodologías de gestión de proyectos ágiles; se describe en qué consisten, qué beneficios aportan frente a las metodologías clásicas y cuáles son las más utilizadas por sus ya contrastados beneficios y el valor que aportan a la gestión de proyectos.

Actualmente existen multitud de aplicaciones creadas para la gestión de proyectos software; cada una de ellas pretende dar solución y facilitar las tareas propias de los gestores y los desarrolladores pertenecientes a los equipos de desarrollo.

Los equipos de desarrollo software suelen estar integrados por gran variedad de recursos, tanto humanos como materiales. Cada uno desempeña una función concreta en el proyecto, pudiendo no tener una dedicación plena al proyecto. Por eso, es necesario que dichos recursos sean compartidos entre la cartera proyectos existentes. Para resolver este planteamiento en las aplicaciones de gestión de proyectos, ha sido requisito fundamental que se puedan gestionar varios proyectos de forma simultánea (gestión multiproyecto), pudiendo repartir la dedicación de los recursos entre los proyectos existentes en la cartera.

En la actualidad, existe un gran número de metodologías de gestión de proyectos, por lo que, en parte, el éxito del proyecto radica en la elección de la más adecuada. Entre todas las metodologías existentes, este estudio se ha centrado en las cada vez más utilizadas metodologías de gestión de proyectos ágiles; se describe en qué consisten, qué beneficios aportan frente a las metodologías clásicas y cuáles son las más utilizadas por sus ya contrastados beneficios y el valor que aportan a la gestión de proyectos. Por lo descrito anteriormente, otro requisito fundamental a la hora de valorar las aplicaciones de gestión de proyectos ha sido la capacidad de soportar y aplicar metodologías ágiles de gestión de proyectos.

En este estudio también se ha tenido en cuenta el tipo de aplicación atendiendo a su instalación y acceso, y se ha realizado la diferenciación entre aplicaciones web- las cuales precisan ser instaladas en un servidor web y son accesibles desde cualquier dispositivo con navegador -, y aplicaciones de escritorio - las cuales precisan estar instaladas en un equipo de forma local y sólo pueden ser accedidas a ellas desde dicho equipo.

En este estudio se han evaluado varias aplicaciones, intentando analizar el cumplimiento de las características comentadas anteriormente, dando como

resultado tres aplicaciones seleccionadas siendo éstas las que pueden aportar más valor a la hora de gestionar una cartera de proyectos.

At present, there are many applications aimed at managing software projects. Every application intends to solve and facilitate tasks to managers and developers belonging to the development teams.

Software development teams are usually made up of many different human and material resources, each of them developing a specific task in the project and sometimes without a full dedication to the project. Therefore, these resources have to be shared within the existing project portfolio. To meet this need in project management applications, the main requirement is to be able to manage several projects simultaneously (multi-project management), thus allowing resources to be shared within the existing project portfolio.

At present, there are a large number of project management methodologies and the success of the project lies in choosing the most appropriate one. Among all the existing methodologies, this study has focused on the increasingly used agile project management methodologies. The study describes the way they work, their added value in comparison traditional methodologies, and which ones are more often used due to their already verified benefits and value in managing projects. Taking into account the above-mentioned characteristics, another key requirement when assessing the project management applications has been their capacity to support and implement project management agile methodologies.

This study has also taken into account the type of application according to its installation and access. A difference is established between web applications – which require to be installed in a web server and are accessible from any device with a web browser – and desktop applications, which must be installed in the equipment to be used and are only accessible from this equipment.

The study has assessed several applications by analyzing the compliance with the above-mentioned characteristics and has chosen three applications that provide the management of the project portfolio with an added value.

Índice

1	Agradecimientos	6
2	Introducción:	8
2.1.	Planteamiento.....	9
2.2.	Objetivos conseguidos.....	10
3	Metodologías de gestión de proyectos	11
3.1.	¿Por qué utilizar metodologías?	13
3.2.	Metodologías clásicas:.....	14
3.2.1.	Inicio:.....	15
3.2.2.	Planificación y diseño:	15
3.2.3.	Ejecución:	17
3.2.4.	Seguimiento y control:	17
3.2.5.	Cierre	19
3.3.	Metodologías Ágiles:	21
3.3.1.	El Manifiesto Ágil:.....	21
3.3.2.	Las Metodologías:.....	24
3.4.	Clásicas Vs Ágiles:	44
3.4.1.	Separación de Diseño y Construcción:.....	44
3.4.2.	La impredecibilidad de los Requisitos:.....	46
3.4.3.	¿Es Imposible la Previsibilidad?	47
3.4.4.	Controlando un Proceso Imprevisible. Iteraciones.....	47
3.4.5.	Orientado a la gente.....	48
3.4.6.	El Cliente Adaptable	49
3.4.7.	Costo de los Cambios en la Construcción de SW	50
3.4.8.	Comparación Ágil - Tradicional.....	52
	Metodología Ágil	52
	Metodología Tradicional	52
4	Gestión Multiproyecto:.....	53
4.1.	Definiciones y objetivos	54

4.2.	El rol de la Oficina de Proyectos	54
4.3.	Procesos de la Oficina de Proyectos	56
5	Tecnologías:	60
5.1.	VMware:	61
5.2.	OpenSuse:	61
5.3.	Windows XP:	63
5.4.	Windows 2003 Server:	63
5.5.	MySQL:	65
5.6.	Apache:	66
5.7.	IIS:	68
5.8.	ASP.NET:	69
5.9.	Java:	70
5.10.	PHP:	71
6	Tipos de aplicaciones de gestión de proyectos	73
6.1.	Aplicaciones de Escritorio:	74
6.2.	Aplicaciones Web:	74
7	Aplicaciones evaluadas:	76
7.1.	OpenProject:	77
7.2.	GanttProject:	80
7.3.	Project Planning and Tracking System (PPTS):	83
7.4.	ScrumWorks:	85
7.4.1.	Entorno Web:	85
7.4.2.	Entorno Escritorio	86
7.5.	JIRA Issue and project tracking:	89
7.6.	DotProject:	92
7.7.	Agilo Trac	94
7.7.1.	Instalación Agilo Trac	97
7.8.	Rational Plan Multiproject:	99
7.8.1.	Instalación de Rational Plan Multiproject:	101
7.9.	Microsoft Project Server 2007	102
7.9.1.	El nivel web de Office Project Server 2007 incluye dos componentes:	103
7.9.2.	El nivel de aplicación de Office Project Server 2007 incluye:	103
7.9.3.	El nivel de base de datos de Office Project Server 2007 incluye las siguientes bases de datos relacionadas de Project Server:	104

7.9.4.	Instalación Microsoft Project Server 2007:	110
8	Bibliografía	117

Índice de ilustraciones

Ilustración 1. Las fases típicas de desarrollo de un proyecto de ingeniería.....	14
Ilustración 2. Grupo de procesos de iniciación.....	15
Ilustración 3. Pasos a seguir para completar el plan de gestión del proyecto...	16
Ilustración 4. Grupo de procesos de ejecución.	17
Ilustración 5. Ciclo de seguimiento y control.	18
Ilustración 6. Grupo de procesos de seguimiento y control.	19
Ilustración 7. Grupo de procesos de control de cierre.	20
Ilustración 8. Proceso definido en XP	25
Ilustración 9. Modelo de caracterización de la Metodología Crystal.	27
Ilustración 10. Ciclo de vida del desarrollo adaptable de software	28
Ilustración 11. Proceso de Scrum	30
Ilustración 12. Ciclo de Desarrollo Manejado por Rasgos.....	31
Ilustración 13. Diagrama de procesos DSDM	33
Ilustración 14. Proceso de pruebas y mejoras de código abierto.	34
Ilustración 15. Proceso de desarrollo de código abierto.....	35
Ilustración 16. Fases, iteraciones y flujos de trabajo de RUP de manera ágil. .	37
Ilustración 17. Función de costos de cambio en el software desarrollado	51
Ilustración 18. PMO tipo gerencial	55
Ilustración 19. Cuota de utilización mundial de sistemas operativos.	65
Ilustración 20. Porcentaje de servidores web usados mundialmente.	68
Ilustración 21. Diagrama de Gannt con OpenProject	77
Ilustración 22. Diagrama de PERT con OpenProject	78
Ilustración 23. Gestión de recursos en OpenProject	78
Ilustración 24. Diagrama de Gannt generado con Gannt Project.....	80
Ilustración 25. Asignación de recursos en GanntProject.	81
Ilustración 26. Diagrama de PERT generado con GanntProject.....	81
Ilustración 27. Planificación de tareas en PPTS.....	83
Ilustración 28. Asignación de recursos en PPTS.....	84
Ilustración 29. Planificación de proyectos en ScrumWorks	85
Ilustración 30. Tareas a realizar en el Spring	86
Ilustración 31. Pila de tareas.....	86
Ilustración 32. Planificación del Spring.....	87

Ilustración 33. Evolución del Spring	87
Ilustración 34. Reportes de evolución del Spring	88
Ilustración 35. Panel individual para la gestión de tareas.....	89
Ilustración 36. Pizarra de reportes.	90
Ilustración 37. Vista de una compañía en DotProject.....	92
Ilustración 38. Vista de proyecto de DotProject.....	93
Ilustración 39. Historias de usuario y tareas.....	94
Ilustración 40. Tablón Kanban.....	95
Ilustración 41. Trabajo pendiente en el Sprint	95
Ilustración 42. Tareas planeadas/cerras y asignación de tareas	96
Ilustración 43. Definición de capacidad de cada recurso	96
Ilustración 44. Sobreasignación de tareas a recursos.....	97
Ilustración 45. Módulos a instalar en Agilo	98
Ilustración 46. Portafolio de proyectos	99
Ilustración 47. Planificación de tareas de un proyecto.	100
Ilustración 48. Gestión de recursos.....	100
Ilustración 49. Arquitectura de Microsoft Project Server.	102
Ilustración 50. Cartera de proyectos activos.	105
Ilustración 51. Planificación de tareas de un proyecto.	106
Ilustración 52. Administración de recursos.....	107
Ilustración 53. Creación de reportes para un proyecto.....	108
Ilustración 54. Panel de comunicación en Project Web Acces.	109
Ilustración 55. Requisitos de instalación.	110
Ilustración 56. Instalación del servidor de aplicaciones.....	111
Ilustración 57. Acceso a Internet Information Services Manager.	112
Ilustración 58. Dar permiso a ISS para que ejecute ASP .NET	112
Ilustración 59. Elección del modo de instalación.....	113
Ilustración 60. Elección del tipo de servidor.	113
Ilustración 61. Instalador de Microsoft Office SharePoint Server.....	114
Ilustración 62. Advertencia instalación instalar Project Web Access.	115
Ilustración 63. Advertencia instalación Project Web Access ActiveX	115

1 **Agradecimientos**

Me gustaría aprovechar estas líneas para agradecer de una manera muy especial, a todas y cada una de las personas que me han apoyado y ayudado a lo largo de toda mi vida universitaria, en la Escuela Universitaria de Ingeniería Técnica de Telecomunicación de la Universidad Politécnica de Madrid.

No puedo nombrarlas a todas ellas, pero sí, hacer una pequeña mención de aquellas que más me han apoyado para poder culminar esta etapa universitaria.

En primer lugar, me gustaría dedicar este trabajo a mi hoy mujer, por su eterna paciencia, comprensión y cariño que ha tenido conmigo a lo largo de nuestras vidas juntos, en todos los años de estudio y de trabajo, por sus noches en vela y los grandes momentos que hemos compartido.

A Blanca, mi hija, que aunque aún sea un pequeño bebé me ha dado las fuerzas para poder finalizar este proyecto y poder emprender nuevos como son la paternidad, con más fuerza y entusiasmo.

A mis padres, por su gran constancia y cariño incondicional, por todas las alegrías y penas que he podido compartir con ellos, por todos los desvelos y sacrificios que han realizado a lo largo de toda su vida por mí, por sus palabras de cariño y su apoyo para afrontar las derrotas con el entusiasmo de poder aprender de ellas.

No sin olvidar a mis hermanos y sobrinos, que me han apoyado siempre e impulsado durante todos estos años de formación.

Para finalizar, pero no menos importantes a mis amigos, a Javier, Daniel y Felipe por todo el ánimo que me han transmitido siempre, por todas las horas que hemos compartido de estudio en las bibliotecas, las noches sin dormir, y los nervios antes de los exámenes, por las alegrías y momentos especiales de nuestras vidas que hemos compartido, por sus apoyos en los momentos malos y su cariño en los momentos felices.

A mis profesores, que me han transmitido las ganas de aprender y todos los conocimientos técnicos para llegar a finalizar esta gran meta. En especial, a Javier Malagón Hernández por aceptar dirigirme este proyecto y todo el esfuerzo que ha depositado en él.

Por todo ello, muchísimas gracias a todos.

2 Introducción:

2.1. Planteamiento

Actualmente existen multitud de aplicaciones creadas para la gestión de proyectos software; cada una de ellas pretende dar solución y facilitar las tareas propias de los gestores y los desarrolladores pertenecientes a los equipos de desarrollo.

Los equipos de desarrollo software suelen estar integrados por gran variedad de recursos, tanto humanos como materiales. Cada uno desempeña una función concreta en el proyecto, pudiendo no tener una dedicación plena al proyecto. Por eso, es necesario que dichos recursos sean compartidos entre la cartera proyectos existentes. Para resolver este planteamiento en las aplicaciones de gestión de proyectos, ha sido requisito fundamental que se puedan gestionar varios proyectos de forma simultánea (gestión multiproyecto), pudiendo repartir la dedicación de los recursos entre los proyectos existentes en la cartera.

En la actualidad, existe un gran número de metodologías de gestión de proyectos, por lo que, en parte, el éxito del proyecto radica en la elección de la más adecuada. Entre todas las metodologías existentes, este estudio se ha centrado en las cada vez más utilizadas metodologías de gestión de proyectos ágiles; se describe en qué consisten, qué beneficios aportan frente a las metodologías clásicas y cuáles son las más utilizadas por sus ya contrastados beneficios y el valor que aportan a la gestión de proyectos. Por lo descrito anteriormente, otro requisito fundamental a la hora de valorar las aplicaciones de gestión de proyectos ha sido la capacidad de soportar y aplicar metodologías ágiles de gestión de proyectos.

2.2. Objetivos conseguidos

En esta memoria se documenta los pasos seguidos para conseguir los objetivos del proyecto. Estos objetivos han sido:

- Estudio sobre las metodologías de gestión de proyectos: Realizado un estudio sobre que son las metodologías ágiles, que nos pueden aportar frente a las metodologías clásicas y la descripción de las principales metodologías ágiles.
- Estudio de la gestión multiproyecto: Realizado un estudio sobre que es la gestión multiproyecto, que objetivos persigue, los roles que intervienen y los procesos que conlleva.
- Evaluación de aplicaciones para la gestión ágil y multiproyecto: Estudio de nueve aplicaciones de gestión de proyectos de diferentes características. Entre dichas aplicaciones se encuentran tanto aplicaciones gratuitas, como de pago y aplicaciones web, de escritorio o de ambos tipos. En dichas aplicaciones se ha evaluado la capacidad de aplicar metodologías ágiles de gestión de proyectos, capacidad de gestionar varios proyectos simultáneamente, compartir recursos entre proyectos, el entorno de trabajo utilizado (web o escritorio), alertas por sobreasignación de recursos, facilidad de manejo, capacidad de exportar o importar proyectos y si son de libre distribución o no. Sobre todas ellas se ha detallado los pros y los contras hallados en función de los aspectos evaluados.
- Selección final de tre aplicaciones catalogadas como más aptas en función de los aspectos descritos anteriormente, dando como resultado:
 - Aplicación Web gratuita. Agilo Trac
 - Aplicación de escritorio gratuita. Rational Plan Multiproject
 - Aplicación Web y de escritorio de pago. Microsoft Project Server 2007

3 Metodologías de gestión de proyectos

Antes de pasar a ver que son las metodologías de gestión de proyectos se va a definir los términos básicos que son necesarios para abordar la gestión de proyectos:

La Real Academia de la Lengua [1] define un proyecto como:

- 1. m. Planta y disposición que se forma para la realización de un tratado, o para la ejecución de algo de importancia.*
- 2. m. Designio o pensamiento de ejecutar algo.*
- 3. m. Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería.*
- 4. m. Primer esquema o plan de cualquier trabajo que se hace a veces como prueba antes de darle la forma definitiva.*

Según la ISO 10.006, un proyecto es:

Proceso único que consiste en un conjunto de actividades coordinadas y controladas, con fecha de inicio y término, que son emprendidas para alcanzar un objetivo, que se establece de acuerdo con requisitos específicos, incluyendo restricciones de plazo, coste y recursos.

Una vez que ya se tiene claro que es un proyecto, podemos definir la metodología como el conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica o en una exposición doctrinal. [2] El término método define el procedimiento empleado para conseguir los objetivos de un proyecto, y la metodología es el estudio del método.

3.1. ¿Por qué utilizar metodologías?

Las metodologías de gestión de proyectos comenzaron a desarrollarse en los años 50, en el seno del ejército estadounidense, intentando reducir el volumen de proyectos que se descontrolaban y pretendiendo solventar problemas comunes que se habían identificado:

- Gran volumen de carga de trabajo planificada o en proceso.
- Costes finales que superan los presupuestos iniciales.
- Problemas en la calidad, valor o utilidad del resultado final.

Las metodologías aportan las siguientes ventajas a la hora de afrontar un proyecto:

- Descripción clara y concisa de los requisitos del proyecto a desarrollar.
- Tener un claro conocimiento de las etapas y pasos a seguir en un proyecto.
- Poder utilizar las técnicas más adecuadas en función del proyecto a desarrollar.
- Tener conocimiento de las pruebas a las que se puede someter el proyecto.
- Optimizar los recursos disponibles en la realización del proyecto.

Como se puede observar, las metodologías aportan muchas ventajas a la hora de desarrollar un proyecto con éxito, pero también se pueden encontrar algunos inconvenientes:

- Puede ser una gran carga de trabajo a desarrollar si el proyecto es sencillo.
- Es necesario dedicar tiempo y recursos al aprendizaje de la metodología.

3.2. Metodologías clásicas:

Las metodologías clásicas o tradicionales realizan un enfoque por etapas, identificando una secuencia de pasos hasta la finalización del proyecto. En el enfoque tradicional de desarrollo de proyectos se pueden distinguir cinco componentes de un proyecto, cuatro etapas más otra etapa de control.

Las fases clásicas de desarrollo de un proyecto de ingeniería son:

- 1 Etapa inicial del proyecto.
- 2 Planificación y diseño del proyecto.
- 3 Proyecto de ejecución y fase de construcción del proyecto.
- 4 Seguimiento del proyecto y sistemas de control.
- 5 Terminación del proyecto.

No todos los proyectos pasan por todas las etapas definidas anteriormente, pudiendo ser terminados antes de llegar a su finalización. Algunos proyectos no siguen una estructura de planificación y/o etapas de seguimiento y pasan a través de los pasos 2, 3 y 4 varias veces.

En el desarrollo de software, este enfoque es conocido como el modelo en cascada, es decir, una serie de tareas tras otras en secuencia lineal. En el desarrollo de software muchas organizaciones se han adaptado al Rational Unified Process (RUP) para adaptarse a esta metodología.

El desarrollo en cascada funciona bien para los proyectos pequeños, bien definidos, pero no en grandes proyectos de carácter indefinido y ambiguo.

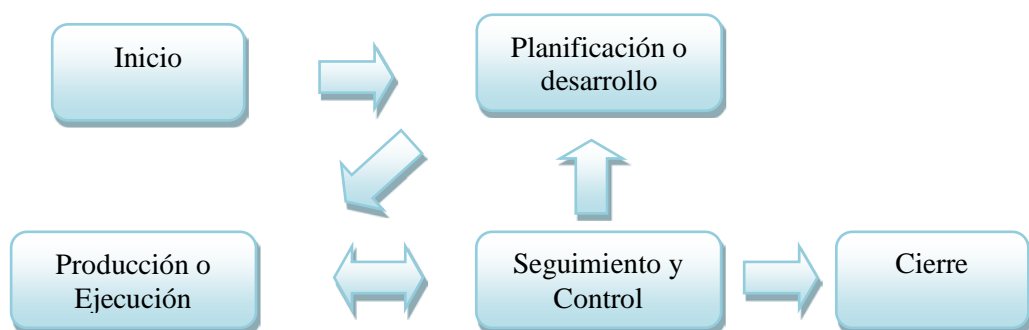


Ilustración 1. Las fases típicas de desarrollo de un proyecto de ingeniería.

3.2.1. Inicio:

Los procesos de iniciación determinan la naturaleza y el alcance del proyecto. [4] Si esta etapa no se realiza bien, es poco probable que el proyecto tenga éxito en satisfacer las necesidades de la empresa. Los controles de los proyectos que se necesitan aquí son la comprensión del ambiente de negocios y asegurarse de que todos los controles necesarios se incorporen en el proyecto. Las deficiencias deben ser reportadas y deben adaptarse las medidas oportunas para corregirlos.

El grupo de procesos de iniciación debe incluir un plan que abarca las siguientes áreas:

- Análisis de la empresa, las necesidades y requisitos de objetivos medibles.
- Revisión de las actuales operaciones.
- Análisis financiero de los costos y beneficios, incluyendo un presupuesto.
- Análisis de los interesados, incluidos los usuarios y personal de apoyo para el proyecto.
- Documentación del proyecto incluyendo los costos, tareas, entregables, y el horario.

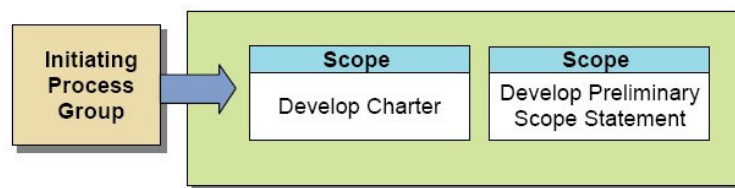


Ilustración 2. Grupo de procesos de iniciación.

3.2.2. Planificación y diseño:

El objetivo principal es planificar el tiempo, el costo y los recursos adecuadamente para estimar el trabajo necesario y para gestionar eficazmente los riesgos durante la ejecución del proyecto. Al igual que con el grupo de procesos de iniciación, la falta de un plan adecuado reduce en gran medida las posibilidades de que el proyecto alcance con éxito sus objetivos.

La planificación de un proyecto en general consta de: [5]

- Determinar la forma del plan (por ejemplo, nivel de detalle).
- El desarrollo de la declaración del alcance del proyecto.
- Seleccionar el equipo de planificación.
- La identificación de las prestaciones y la creación de la estructura de división del trabajo.
- Identificar las actividades necesarias para completar las entregas y las redes de las actividades en su secuencia lógica.
- Estimación de la necesidad de recursos para las actividades.
- Estimar el tiempo y costo de las actividades.
- El desarrollo del calendario.
- La elaboración del presupuesto.
- Riesgo de la planificación.
- Obtener la aprobación formal para comenzar a trabajar.

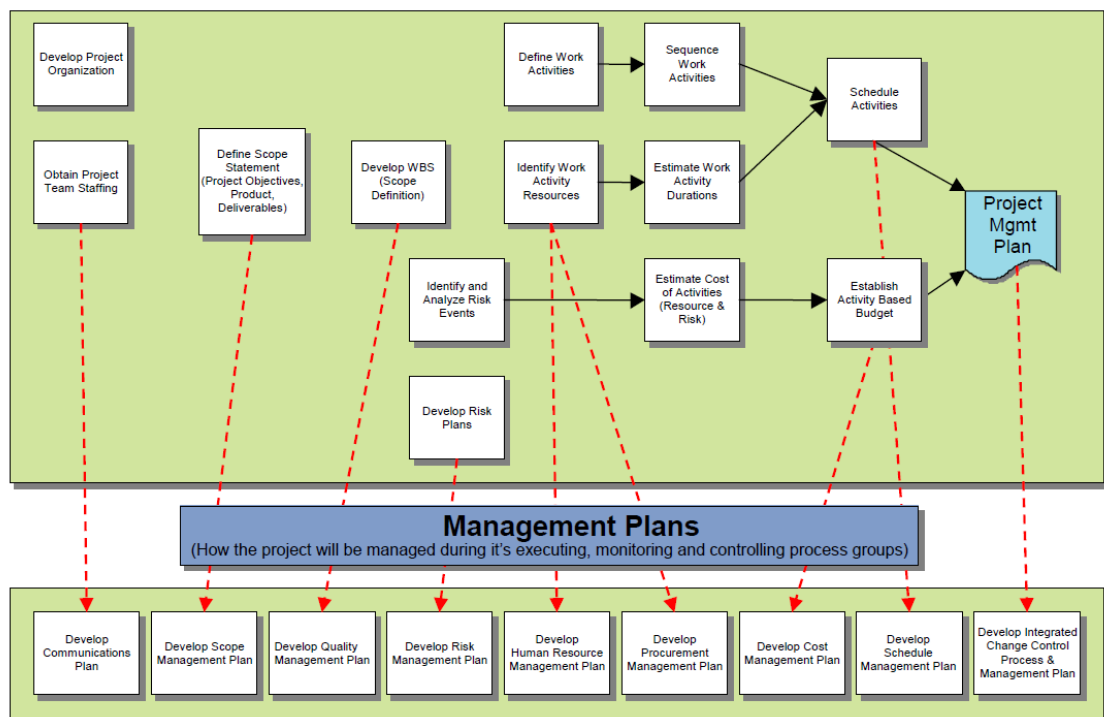


Ilustración 3. Actividades y pasos a seguir para completar el plan de gestión del proyecto.

Existen procesos adicionales, tales como la planificación de las comunicaciones y de gestión del alcance, las funciones y las responsabilidades, la determinación de qué comprar para el proyecto y la

celebración de una reunión de lanzamiento se aconsejan realizarlas en este proceso.

Para el desarrollo de nuevos proyectos, el diseño conceptual del producto final puede realizarse simultáneamente con las actividades de planificación del proyecto, y puede ayudar a informar al equipo de planificación en la identificación de resultados y actividades de planificación.

3.2.3. Ejecución:

El proceso de ejecución se compone de los pasos necesarios para completar el trabajo definido en el plan del proyecto y así cumplir los requisitos del proyecto. Durante el proceso de ejecución se deben coordinar los recursos, así como integrar y realizar las actividades del proyecto, conforme al plan de gestión de proyectos. Los productos son el resultado de los procesos llevados a cabo, tal y como se define en el plan de gestión de proyectos.

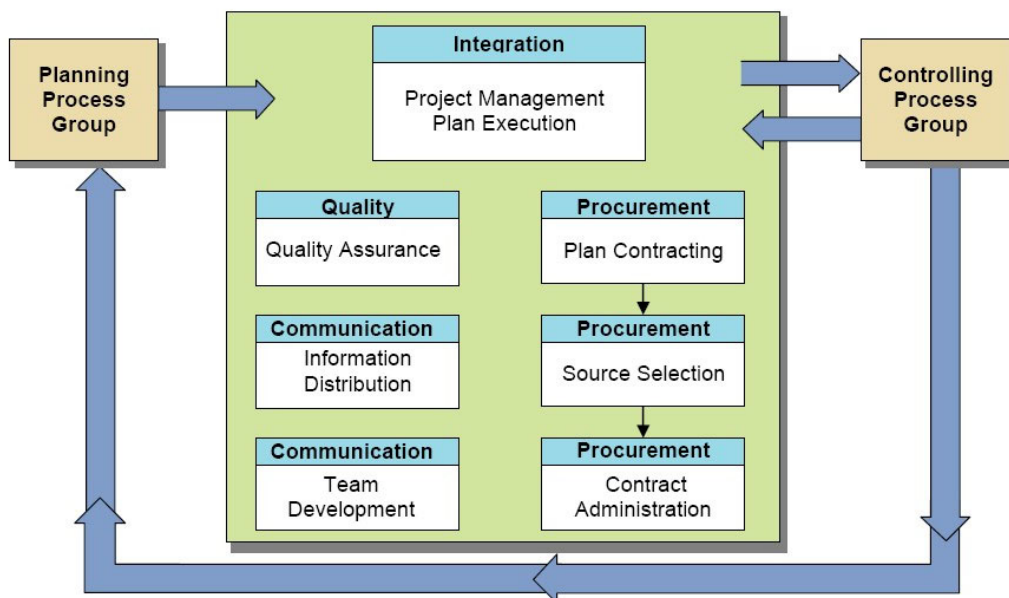


Ilustración 4. Grupo de procesos de ejecución.

3.2.4. Seguimiento y control:

La fase de seguimiento y control se compone de aquellos procesos realizados para observar la ejecución del proyecto y para que los problemas potenciales se puedan identificar y se puedan adoptar las medidas correctivas oportunas cuando sea necesario, y así controlar la ejecución del proyecto. La ventaja fundamental es que los resultados del proyecto se observan y se miden

regularmente identificando las variaciones producidas del plan de gestión de proyectos.

Seguimiento y Control incluye las siguientes tareas: [6]

- La medición de las actividades del proyecto en curso ("dónde estamos").
- Monitoreo de las variables del proyecto (costo, esfuerzo, alcance, etc.) contra el plan de gestión de proyectos y la línea base de rendimiento del proyecto.
- Identificar medidas correctivas para abordar los problemas y los riesgos.

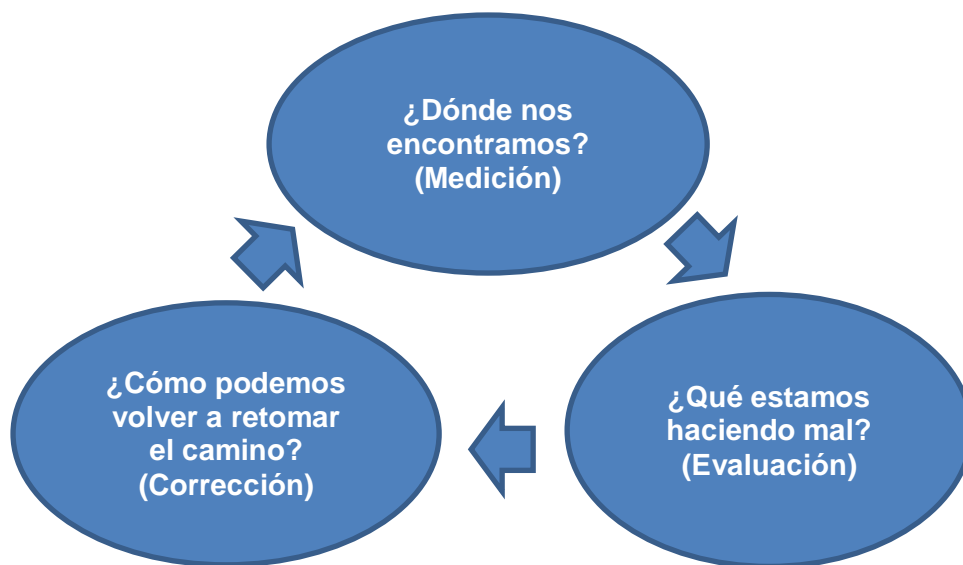


Ilustración 5. Ciclo de seguimiento y control.

En proyectos de fases múltiples, el proceso de supervisión y control también proporciona retroalimentación entre las fases del proyecto, con el fin de implementar acciones correctivas o preventivas para que el proyecto cumpla con el plan de gestión de proyectos.

El mantenimiento del proyecto es un proceso continuo que debe incluir: [7]

- El apoyo continuo de los usuarios finales.
- Corrección de errores.
- Actualizaciones del software a través del tiempo.

En esta etapa, los auditores deben prestar atención a la eficacia y rapidez con la que los problemas son resueltos.

En el transcurso de cualquier proyecto de desarrollo de software, el alcance del trabajo puede cambiar. El cambio es una parte normal del proceso de desarrollo de software. Los cambios pueden ser el resultado de modificaciones de diseño necesarias, la disponibilidad de material, los cambios solicitados por el contratista, ingeniería de valor, los efectos de terceros, etc. Más allá de la ejecución del cambio en el desarrollo, el cambio normalmente necesita ser documentado para mostrar lo que se desarrolló en realidad. Esto se conoce como el cambio de gestión. Por lo tanto, el propietario por lo general requiere un registro final para mostrar todos los cambios o, más específicamente, cualquier cambio que modifique las partes tangibles del software terminado. Por norma general, el registro se hace en los documentos del contrato. Cuando se introducen cambios en el proyecto, la viabilidad del proyecto tiene que ser re-evaluada. Es importante no perder de vista los objetivos iniciales y los objetivos del proyecto. Cuando los cambios se acumulan, el resultado previsto no puede justificar la inversión original propuesta en el proyecto.

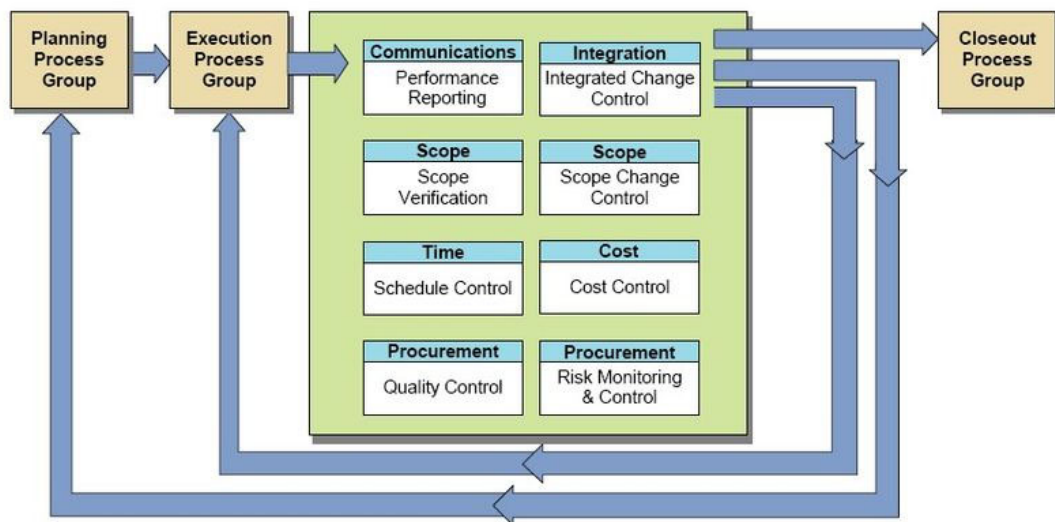


Ilustración 6. Grupo de procesos de seguimiento y control.

3.2.5. Cierre

El cierre incluye la aceptación formal del proyecto. También se incluyen las actividades administrativas pendientes y las tareas de documentar todo lo aprendido durante el desarrollo del proyecto. Esta fase consiste en: [7]

- **Cierre del proyecto:** Finalizar todas las actividades en todos los grupos de proceso para cerrar formalmente el proyecto o una fase del proyecto.

- **Cierre del contrato:** Completar y cerrar cada contrato aplicable al proyecto o fase del proyecto.



Ilustración 7. Grupo de procesos de control de cierre.

3.3. Metodologías Ágiles:

En febrero de 2001, tras una reunión celebrada en Utah, Estados Unidos, nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras dicha reunión se creó The Agile Alliance¹, una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones a que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, que es un documento que resume la filosofía ágil.

3.3.1. El Manifiesto Ágil:

Según el Manifiesto se valora: [8]

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- **Desarrollar software que funciona más que conseguir una buena documentación.** La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión

¹ <http://www.agilealliance.org/>

importante. Estos documentos deben ser cortos y centrarse en lo fundamental.

- **La colaboración con el cliente más que la negociación de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito. Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores son la base de los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional o clásico. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo.

Los principios son: [8]

1. *La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.*
2. *Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
3. *Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.*
4. *La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*
5. *Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*
6. *El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*

7. *El software que funciona es la medida principal de progreso.*
8. *Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*
9. *La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*
10. *La simplicidad es esencial.*
11. *Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*
12. *En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, según esto ajusta su comportamiento.*

3.3.2. Las Metodologías:

Varias metodologías encajan bajo el estandarte de ágil. Mientras todas ellas comparten muchas características, también hay algunas diferencias significativas.

3.3.2.1. Programación Extrema (XP):

De todas las metodologías ágiles, ésta es la que ha recibido más atención. De alguna manera la popularidad de XP se ha vuelto un problema, pues ha acaparado la atención fuera de las otras metodologías y sus valiosas ideas.

Las raíces de XP nacen en la comunidad de Smalltalk, y en particular en la colaboración cercana de Kent Beck y Ward Cunningham a finales de los '80. Ambos refinaron sus prácticas en numerosos proyectos a principios de los '90, extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente.

El paso de la práctica informal a una metodología sucedió en la primavera de 1996. A Kent Beck se le pidió revisar el progreso del proyecto denominado C3 (Chrysler Comprehensive Compensation System) para Chrysler. El proyecto estaba llevado en Smalltalk por una compañía contratista, y estaba teniendo problemas, debido a la baja calidad de su código base. Kent Beck recomendó deshacerse del código en su totalidad y empezar desde el principio. El proyecto comenzó bajo su dirección y el proyecto se volvió el buque insignia y el campo de entrenamiento de XP.

La primera fase del C3 fue muy exitosa y comenzó a principios de 1997. El proyecto continuó desde entonces y después se encontró con dificultades, lo que derivó en la cancelación del desarrollo en 1999, lo que denota que la Programación Extrema no es garantía de éxito.

La metodología XP se basa en cuatro valores: [9]

- Comunicación.
- Retroalimentación.
- Simplicidad.
- Coraje.

Construye sobre ellos una docena de prácticas que los proyectos XP deben seguir. Muchas de estas prácticas son técnicas antiguas, tratadas y probadas. Además de resucitar estas técnicas, XP las une en un todo, donde cada una refuerza a las demás.

Una de las más llamativas, así como inicialmente atractiva, es su fuerte énfasis en las pruebas. Mientras todas las metodologías de gestión de proyectos mencionan la comprobación, la mayoría lo hace con muy poco énfasis. Sin embargo XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción, lo que rinde una plataforma altamente estable para el desarrollo futuro.

En XP se construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado que combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada de entre todas las metodologías adaptables.

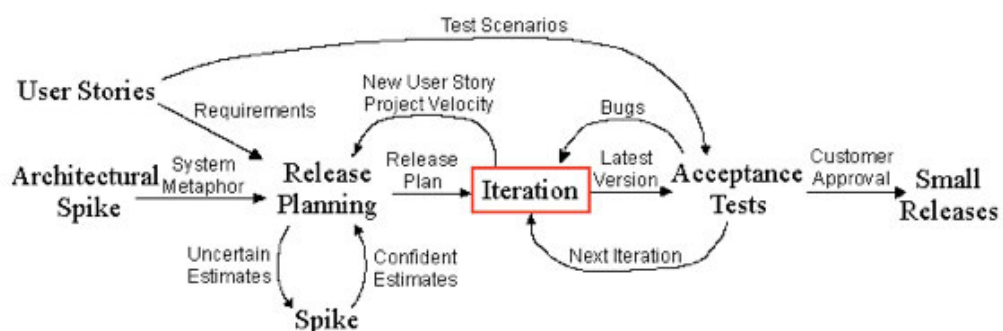


Ilustración 8. Proceso definido en XP

XP ha desarrollado un liderazgo amplio, muchos de ellos provenientes del proyecto fundamental C3. Como resultado hay muchas fuentes para más información. Kent Beck escribió "Extreme Programming Explained" [9], el manifiesto clave de XP que explica las razones detrás de la metodología y es bastante amplia como para que la gente pueda saber si se interesan en seguirla. En el último par de años se han publicado gran cantidad de libros sobre XP, de los cuales son bastante similares en que describen el proceso entero desde el punto de vista de varios seguidores.

También hay otras referencias online en las cuales se puede ampliar los conocimientos sobre XP, como son <http://www.programacionextrema.org> y un Grupo de Discusión en Yahoo en <http://groups.yahoo.com/group/xpspanish>.

3.3.2.2. La Familia de Crystal de Cockburn:

Alistair Cockburn ha estado trabajando en metodologías desde que IBM le encargó escribir sobre metodologías a inicios de los '90. No obstante, su acercamiento no es como la mayoría de los metodologistas. En lugar de partir solamente de su experiencia personal para construir una teoría de cómo deben hacerse las cosas, él complementa su experiencia directa con la búsqueda activa de proyectos y ver cómo trabajan, además de alterar sus puntos de vista con sus descubrimientos.

Su libro, "Surviving Object-Oriented Projects: a Manager's Guide" [10], fue su primer consejo en proyectos, y sigue siendo una primera recomendación de libro para ejecutar proyectos iterativos. Más recientemente Alistair Cockburn escribió un libro de apreciación de "Agile Software Development" [11] que observa los principios subyacentes de este tipo de metodologías.

Desde ese libro él ha explorado más los métodos ágiles, trabajando con la familia de metodologías Cristal. Es una familia porque él cree que los tipos diferentes de proyectos requieren tipos diferentes de metodologías.

Alistair Cockburn mira esta variación a lo largo de dos ejes:

- El número de personas en el proyecto.
- Las consecuencias de los errores.

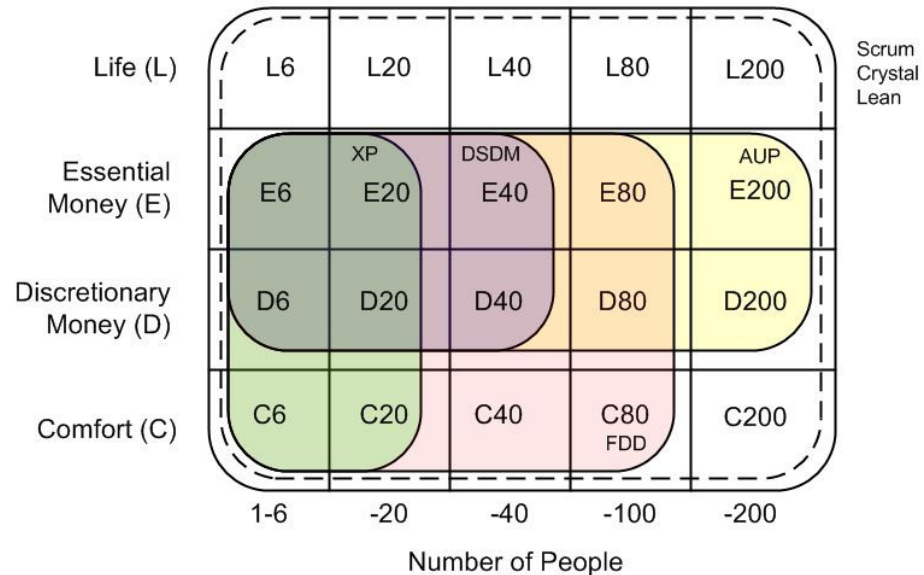


Ilustración 9. Modelo de caracterización de la Metodología Crystal.

Cada metodología encaja en una parte diferente de la rejilla, de modo que para un proyecto de 40 personas que puede perder dinero discrecionalmente tiene una metodología diferente a la de un proyecto vital de seis personas.

Crystal comparte con XP una orientación humana, pero esta centralización en la gente se hace de una manera diferente. Alistair Cockburn considera que las personas encuentran difícil seguir un proceso disciplinado, así que más que seguir la alta disciplina de XP, explora la metodología menos disciplinada que aún podría tener éxito, intercambiando conscientemente productividad por facilidad de ejecución. Él considera que aunque Crystal es menos productivo que XP, más personas serán capaces de seguirlo.

Alistair Cockburn también pone mucho peso en las revisiones al final de la iteración, animando al proceso a ser auto mejorable. Su aserción es que el desarrollo iterativo está para encontrar los problemas lo antes posible, y entonces permitir corregirlos. Esto pone más énfasis en la gente supervisando su proceso y afinándolo a medida que desarrollan.

3.3.2.3. El Desarrollo de Software Adaptable (ASD) de Highsmith

Jim Highsmith ha pasado muchos años trabajando con metodologías clásicas. Él las desarrolló, instaló, enseñó, y llegó a la conclusión de que son profundamente defectuosas, sobre todo para los negocios modernos.

Su libro “Adaptive Software Development” [12] se enfoca en la naturaleza adaptable de las nuevas metodologías, poniendo un especial énfasis en aplicar las ideas que se originaron en el mundo de los sistemas complejos adaptables (teoría del caos). No proporciona el tipo de prácticas detalladas como lo hace XP, pero si proporciona la base fundamental de por qué el desarrollo adaptable es importante y las consecuencias a todos los niveles de la organización y la gerencia.

En ASD hay tres fases solapadas, no lineales: especulación, colaboración, y aprendizaje.

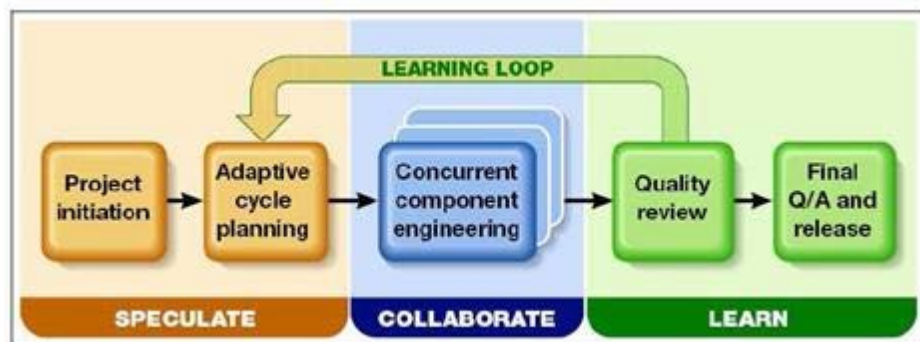


Ilustración 10. Actividades del ciclo de vida del desarrollo adaptable de software

Jim Highsmith ve la planificación como un ambiente al que hay que ir adaptándose, ya que los resultados son naturalmente imprevisibles. En la planificación tradicional, las desviaciones del plan son errores que deben corregirse. En un ambiente adaptable las desviaciones son guía hacia la solución correcta.

En este ambiente imprevisible se necesita que las personas colaboren de la mejor manera para tratar con la incertidumbre. La atención de la gerencia es menor en lo que tiene que hacer la gente, y mayor sobre la comunicación para que las personas puedan proponer las respuestas creativas ellos mismos.

En ambientes predictivos, el aprendizaje se desalienta a menudo. Las cosas se ponen de antemano y entonces se sigue ese diseño. En un ambiente adaptable, aprender desafía a todos (desarrolladores y sus clientes) a examinar sus presunciones y usar los resultados de cada ciclo de desarrollo para adaptar el siguiente. El aprendizaje como tal es un rasgo continuo e importante, que asume que los planes y los diseños deben cambiar conforme avanza el desarrollo.

El beneficio del Ciclo de Vida de Desarrollo Adaptable es que obliga a confrontar los modelos mentales que están en la raíz del autoengaño de las personas. Obliga a las personas a estimar con realismo su habilidad. Con este énfasis, el trabajo de Jim Highsmith se enfoca directamente en fomentar las partes difíciles del desarrollo adaptable, en particular cómo fomentar la colaboración y el aprendizaje dentro del proyecto. Como tal, su libro [12] ayuda a dar ideas para fomentar estas áreas "suaves" que hacen un buen complemento a los acercamientos basados en una práctica aterrizada como XP, FDD y Crystal.

3.3.2.4. **Scrum:**

La metodología ágil Scrum se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles.

Scrum divide un proyecto en iteraciones (que se llaman carreras cortas o sprint) de unos 15 a 30 días. Antes de que comience un sprint se define la funcionalidad requerida para ese sprint y entonces se deja al equipo para que la entregue. El punto es estabilizar los requisitos durante el sprint.

Sin embargo la gerencia no se desentiende durante el sprint. Todos los días el equipo sostiene una reunión corta (quince minutos), llamada daily, dónde el equipo discurre lo que hará al día siguiente. En particular muestran a los bloques de la gerencia: los impedimentos para progresar que se atraviesan y que la gerencia debe resolver. También informan lo que se ha hecho para que la gerencia tenga una actualización diaria de cómo va el proyecto.

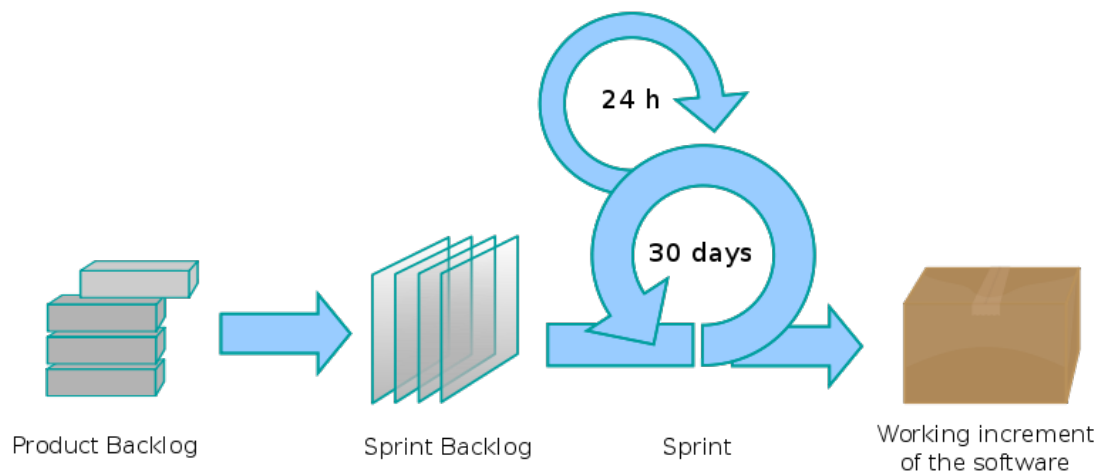


Ilustración 11. Proceso de Scrum

La literatura de Scrum se enfoca principalmente en la planeación iterativa y el seguimiento del proceso. Es muy cercana a las otras metodologías ágiles en muchos aspectos y debe funcionar bien con las prácticas de código de XP.

Después de mucho tiempo sin un libro, finalmente Ken Schwaber y Mike Beedle escribieron el primer libro de Scrum que lo llamaron “Agile Software Development with Scrum” [13]. Probablemente la mejor apreciación global sobre Scrum está en el sitio <http://www.controlchaos.com> de Ken Schwaber. Hay también una buena apreciación global de las prácticas de Scrum en el libro de Neil Harrison, Brian Foote, Hans Rohnert titulado “Pattern Languages of Program Design 4 (Software Patterns Series)” [14].

3.3.2.5. Desarrollo Manejado por Rasgos (FDD)

El Desarrollo Manejado por Rasgos (FDD por sus siglas en inglés de Feature-Driven Development) fue desarrollado por Jeff De Luca y el gurú de la Orientación a Objetos Peter Coad. Como las otras metodologías adaptables, se enfoca en iteraciones cortas que entregan funcionalidad tangible. En el caso del FDD las iteraciones duran dos semanas.

El FDD tiene cinco procesos. Los primeros tres se hacen al principio del proyecto y los dos últimos se hacen en cada iteración.

- Desarrollar un Modelo Global
- Construir una Lista de los Rasgos
- Planear por Rasgo
- Diseñar por Rasgo
- Construir por Rasgo

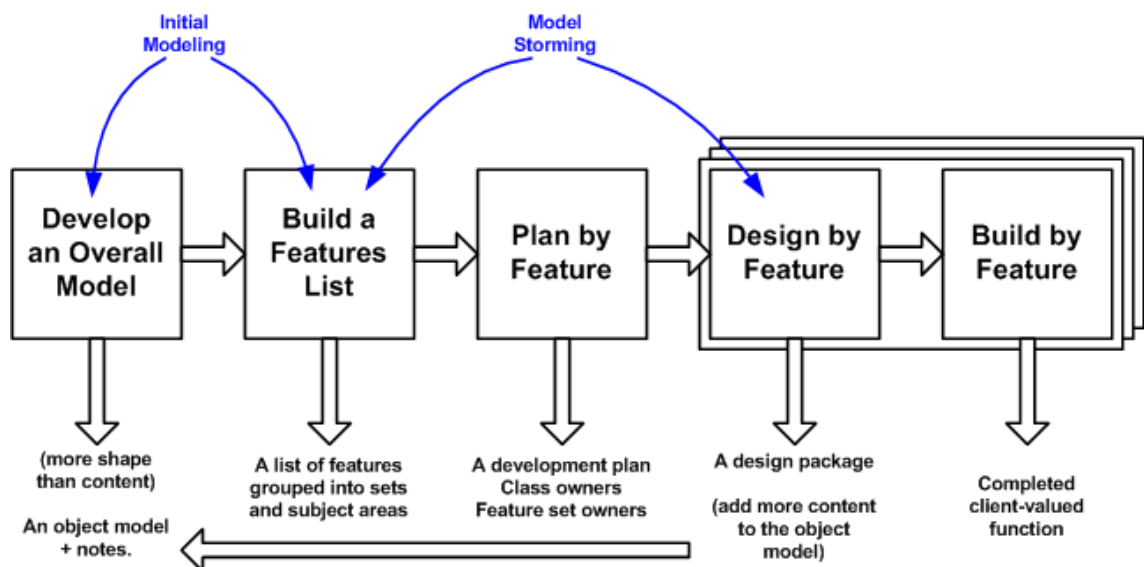


Ilustración 12. Ciclo de Desarrollo Manejado por Rasgos

Cada proceso se divide en tareas y se da un criterio de comprobación.

Los desarrolladores entran en dos tipos:

- Dueños de clases
- Programadores jefe.

Los programadores jefe son los desarrolladores más experimentados. A ellos se les asignan rasgos a construir. Sin embargo ellos no los construyen solos. Sólo identifican qué clases se involucran en la implantación de un rasgo y juntan a los dueños de dichas clases para que formen un equipo para desarrollar ese rasgo. El programador jefe actúa como el coordinador, diseñador líder y mentor, mientras los dueños de clases hacen gran parte de la codificación del rasgo.

La documentación sobre FDD era muy escasa, pero finalmente hay un libro completo sobre FDD titulado “A Practical Guide to Feature-Driven Development (The Coad Series)” [15] de la autoría de Stephen R Palmer y John M. Felsing. Jeff De Luca, el inventor primario, ya tiene un portal FDD en la URL <http://www.featuredrivendevelopment.com>, con artículos, blogs y foros de discusión. La descripción original está en el libro “Java Modeling In Color With UML” [16].

3.3.2.6. Método de Desarrollo de Sistema Dinámico (DSDM)

El DSDM (Dynamic Systems Development Method) empezó en Gran Bretaña en 1994 como un consorcio de compañías del Reino Unido que querían construir sobre RAD (Rapid Applications Development) Desarrollo Rápido de Aplicaciones y desarrollo iterativo. Habiendo empezado con 17 fundadores ahora tiene más de mil miembros y ha crecido fuera de sus raíces británicas. Siendo desarrollado por un consorcio, tiene un sabor diferente a muchos de los otros métodos ágiles. Tiene una organización que la apoya con manuales, cursos de entrenamiento, programas de certificación y demás, pero todos ellos con precio, por lo cual ha limitado la investigación popular sobre su metodología. Sin embargo, Jennifer Stapleton ha escrito el libro “DSDM” [17] que da una apreciación global de la metodología. El método empieza con un estudio de viabilidad y negocio. El estudio de viabilidad considera si DSDM es apropiado para el proyecto. El estudio de negocio es una serie corta de talleres para entender el área de negocio donde tiene lugar el desarrollo. También propone esbozos de arquitecturas del sistema y un plan del proyecto.

El resto del proceso forma tres ciclos entretejidos:

- El ciclo del modelo funcional produce documentación de análisis y prototipos.
- El ciclo de diseño del modelo diseña el sistema para uso operacional.
- El ciclo de implantación se ocupa del despliegue al uso operacional.

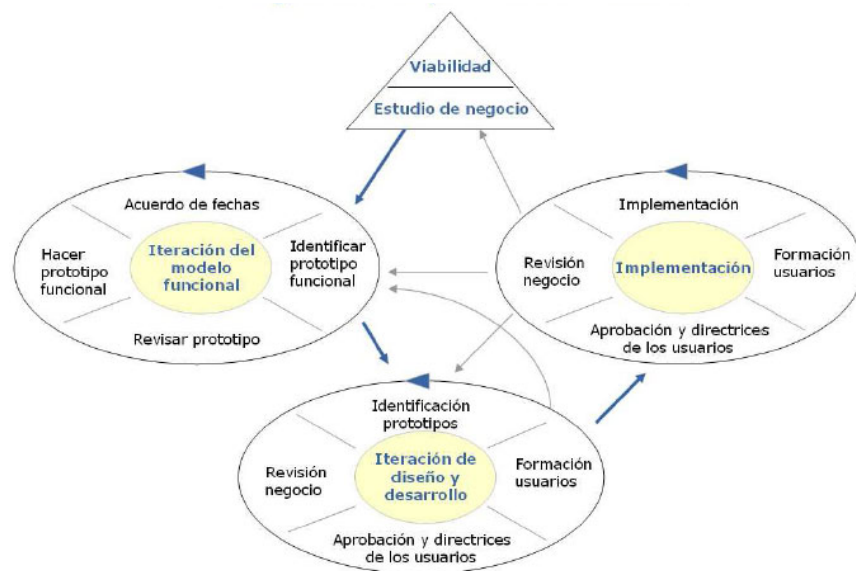


Ilustración 13. Diagrama de procesos DSDM

DSDM tiene principios subyacentes que incluyen una interacción activa del usuario, entregas frecuentes, equipos autorizados, pruebas a lo largo del ciclo. Igual que otros métodos ágiles, usan ciclos de plazos cortos de entre dos y seis semanas. Hay un énfasis en la alta calidad y adaptabilidad hacia requisitos cambiantes.

3.3.2.7. Software de Código Abierto (OSS)

Este título podría sorprender, ya que el código abierto (OS por su sigla en inglés de Open Source) es un estilo de software, no tanto un proceso. Sin embargo hay una manera definida de hacer las cosas en la comunidad de código abierto, y mucho de su acercamiento es tan aplicable a los proyectos de código cerrado como a los de código abierto. En particular su proceso engrana equipos de trabajo físicamente distribuidos, lo que es importante porque la

mayoría de los procesos adaptables exigen equipos locales.

La mayoría de los proyectos de código abierto tienen uno o más mantenedores. Un mantenedor es la única persona a la que se le permite integrar un cambio en el almacén de código fuente. Sin embargo otras personas pueden hacer cambios a la base del código. La diferencia importante es que estas otras personas necesitan enviar su cambio al mantenedor que entonces lo revisa y lo aplica a la base del código. Normalmente estos cambios son hechos en forma de archivos de parches que hacen este proceso más fácil. Así, el mantenedor es responsable de coordinar los parches y mantener la cohesión en el diseño del software.

Proyectos diferentes manejan el papel del mantenedor de diferentes maneras. Algunos tienen un mantenedor para el proyecto entero, algunos lo dividen en módulos y tiene un mantenedor por módulo, algunos rotan el mantenedor, algunos tienen múltiples mantenedores sobre el mismo código, otros tienen una combinación de estas ideas. La mayor parte de la gente de código abierto son de tiempo parcial, así que hay una duda en qué tan bien se coordina un equipo así para un proyecto de tiempo completo.

Un rasgo particular del desarrollo de código abierto es que la depuración se puede realizar de manera paralela, por lo que muchas personas pueden involucrarse en el depurado. Cuando encuentran un defecto pueden enviar el parche al mantenedor. Esto es un buen papel para los no mantenedores ya que la mayor parte del tiempo se gasta en encontrar bugs. También es bueno para gente sin mucha destreza en programación.

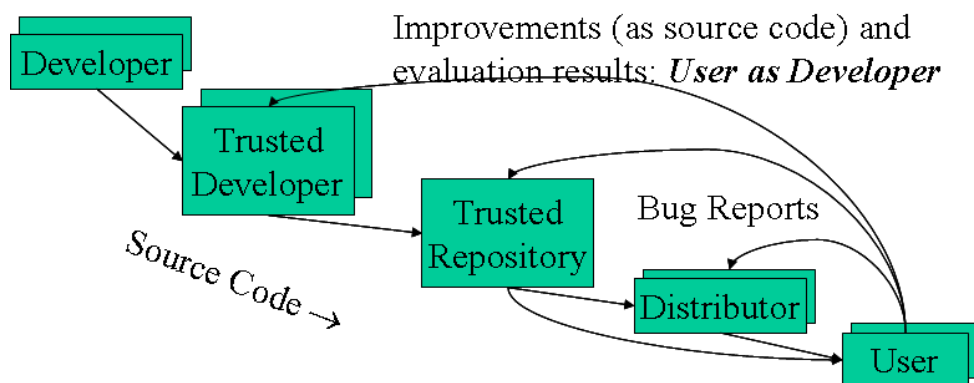


Ilustración 14. Proceso de pruebas y mejoras de código abierto.

El proceso para el código abierto aún no se ha documentado bien. La referencia más famosa es el artículo de Eric Raymond “La Catedral y el Bazar”, que aunque es una descripción excelente también es bastante informal. La publicación de Karl Fogel sobre “Desarrollo de Código Abierto con CVS” [18], también contiene varios buenos capítulos sobre el proceso de código abierto que incluso serían interesantes para aquellos que no quieren usar el CVS. CVS son las siglas de Concurrent Versions System (Sistema de Versiones Concurrentes), un sistema cliente-servidor que permite a los desarrolladores realizar el seguimiento de las diferentes versiones del código fuente de un proyecto. El CVS es especialmente útil cuándo más de una persona trabaja sobre un archivo específico. En tales situaciones, es posible que un desarrollador sobrescriba accidentalmente los cambios que ha realizado otro desarrollador. CVS resuelve este problema haciendo que cada desarrollador trabaje sobre su propia copia del código (lo que se denomina *workspace* o espacio de trabajo) y permitiendo posteriormente unir los cambios de todos los desarrolladores en un repositorio común [19].

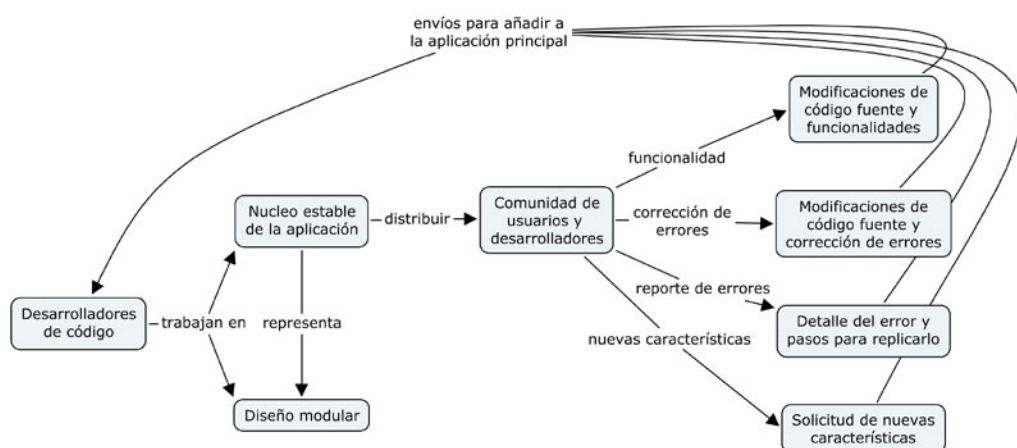


Ilustración 15. Proceso de desarrollo de código abierto.

3.3.2.8. Rational Unified Process (RUP):

Siempre que se discuten sobre métodos orientados a objetos, inevitablemente se llega a Rational Unified Process. El Proceso Unificado fue desarrollado como el proceso complementario al UML². El RUP es un almacén de proceso y como tal puede acomodar una gran variedad de procesos. De hecho ésta es la crítica principal al RUP por parte de algunos autores: como puede ser cualquier cosa acaba siendo nada. Prefieren un proceso que diga qué hacer en lugar de dar opciones infinitas.

Craig Larman es un fuerte defensor de usar el RUP de una manera ágil. Su libro “Applying UML and Patterns” [20] sobre desarrollo orientado a objetos contiene un proceso que está muy basado en su pensamiento ligero del RUP. Su visión es que mucho del reciente empujón hacia los métodos ágiles no es nada más que aceptar desarrollo orientado a objetos de la corriente principal que ha sido capturada como RUP. Una de las cosas que hace Craig Larman es pasarse los primeros dos o tres días de una iteración mensual con todo el equipo usando el UML para perfilar el diseño del trabajo a hacerse durante la iteración. Esto no es una norma de la que no pueda desviarse, sino un boceto que da una perspectiva sobre cómo pueden hacerse las cosas en la iteración.

² UML / Unified Modeling Language): es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad: Está respaldado por el Object Management Group. y puede considerarse un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

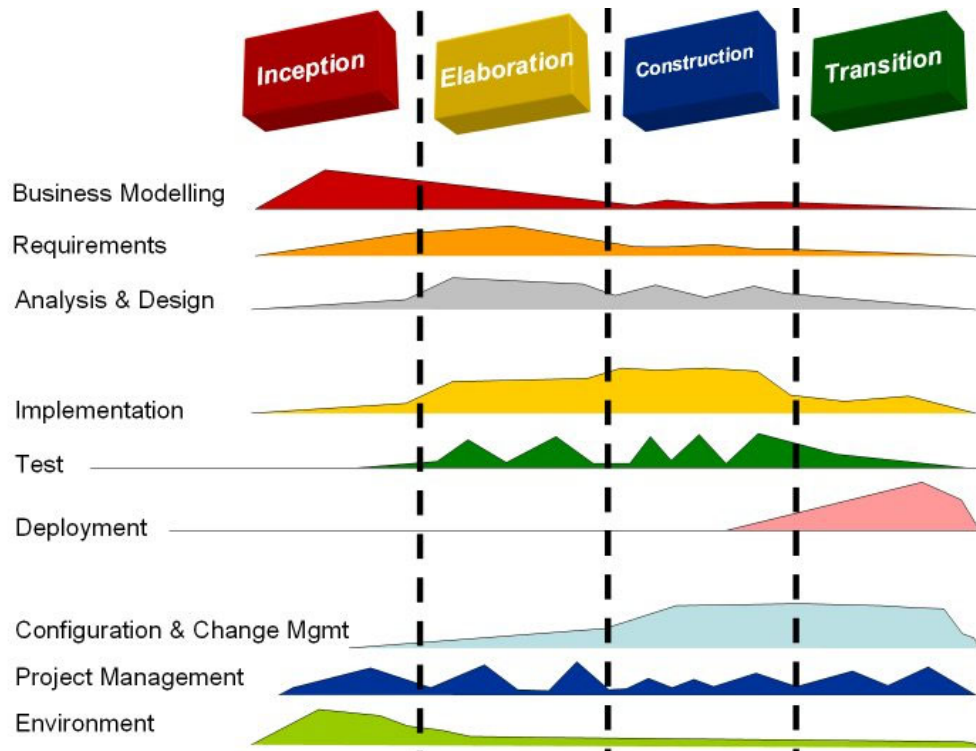


Ilustración 16. Fases, iteraciones y flujos de trabajo de RUP de manera ágil.

Según los críticos, una de las cosas claves que necesita el RUP es que los líderes del RUP en la industria enfatizen su acercamiento al desarrollo de software. Más de una vez se oye a la gente que usa el RUP hablando que están usando un proceso de desarrollo estilo cascada. Philippe Kruchten y su equipo son firmes creyentes en el desarrollo iterativo. Clarificando estos principios y animando las versiones ágiles del RUP tales como los trabajos de Craig Larman y de Robert Martin tendrá un efecto importante.

3.3.2.9. Agile Modeling

Agile Modeling (AM) fue propuesto por Scott Ambler [21] no tanto como un método ágil cerrado en sí mismo, sino como complemento de otras metodologías, sean éstas ágiles o convencionales. En el caso de XP, los practicantes podrían definir mejor los procesos de modelado que en ellos faltan, y en el caso de RUP el modelado ágil permite hacer más ligeros los procesos que ya usan. AM es una estrategia de modelado (de clases, de datos, de procesos) pensada para contrarrestar la sospecha de que los métodos ágiles no modelan y no documentan. Se lo podría definir como un proceso de

software basado en prácticas cuyo objetivo es orientar el modelado de una manera efectiva y ágil.

Los principales objetivos de AM son:

1. Definir y mostrar de qué manera se debe poner en práctica una colección de valores, principios y prácticas que conducen al modelado de peso ligero.
2. Enfrentar el problema de la aplicación de técnicas de modelado en procesos de desarrollo ágiles.
3. Enfrentar el problema de la aplicación de las técnicas de modelado independientemente del proceso de software que se utilice.

Los valores de AM incluyen a los de XP: comunicación, simplicidad, retroalimentación y coraje, añadiendo humildad. Una de las mejores caracterizaciones de los principios subyacentes a AM está en la definición de sus alcances:

1. AM es una actitud, no un proceso prescriptivo. Comprende una colección de valores a los que los modeladores ágiles adhieren principios en los que creen y prácticas que aplican. Describe un estilo de modelado.
2. AM es suplemento de otros métodos. El primer foco es el modelado y el segundo la documentación.
3. AM es una tarea de conjunto de los participantes. No hay “yo” en AM.
4. La prioridad es la efectividad. AM ayuda a crear un modelo o proceso cuando se tiene un propósito claro y se comprenden las necesidades de la audiencia; contribuye a aplicar los artefactos correctos para afrontar la situación inmediata y a crear los modelos más simples que sea posible.

5. AM es algo que funciona en la práctica, no una teoría académica. Las prácticas han sido discutidas desde 2001 en la comunidad <http://www.agilemodeling.com/feedback.htm>
6. AM no es una bala de plata.
7. AM es para el programador promedio, pero no reemplaza a la gente competente.
8. AM no es un ataque a la documentación. La documentación debe ser mínima y relevante.
9. AM no es un ataque a las herramientas CASE³.
10. AM no es para cualquiera.

Los principios de AM especificados por Scott Ambler incluyen:

1. **Presuponer simplicidad.** La solución más simple es la mejor.
2. **El contenido es más importante que la representación.** Pueden ser notas, pizarras o documentos formales. Lo que importa no es el soporte físico o la técnica de representación, sino el contenido.
3. **Abrazar el cambio.** Aceptar que los requerimientos cambian.
4. **Habilitar el esfuerzo siguiente.** Garantizar que el sistema es suficientemente robusto para admitir mejoras posteriores. Debe ser un objetivo, pero no el primordial.
5. **Todo el mundo puede aprender de algún otro.** Reconocer que nunca se domina realmente algo.
6. **Cambio incremental.** No esperar hacerlo bien la primera vez.

³ CASE: Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

7. **Conocer tus modelos.** Saber cuáles son sus fuerzas y sus debilidades.
8. **Adaptación local.** Producir sólo el modelo que resulte suficiente para el propósito.
9. **Maximizar la inversión del cliente.**
10. **Modelar con un propósito.** Si no se puede identificar para qué se está haciendo algo ¿para qué molestarse?
11. **Modelos múltiples.** Múltiples paradigmas en convivencia, según se requiera.
12. **Comunicación abierta y honesta.**
13. **Trabajo de calidad.**
14. **Realimentación rápida.** No esperar que sea demasiado tarde.
15. **El software es el objetivo primario.** Debe ser de alta calidad y coincidir con lo que el usuario espera.
16. **Viajar ligero de equipaje.** No crear más modelos de los necesarios.
17. **Trabajar con los instintos de la gente.**

Lo más concreto de AM es su rico conjunto de prácticas [22], cada una de las cuales se asocia a lineamientos decididamente narrativos, articulados con minuciosidad, pero muy lejos de los rigores cuantitativos:

1. Colaboración activa de los participantes.
2. Aplicación de estándares de modelado.
3. Aplicación adecuada de patrones de modelado.
4. Aplicación de los artefactos correctos.
5. Propiedad colectiva de todos los elementos.
6. Considerar la verificabilidad.
7. Crear diversos modelos en paralelo.

8. Crear contenido simple.
9. Diseñar modelos de manera simple.
10. Descartar los modelos temporales.
11. Exhibir públicamente los modelos.
12. Formalizar modelos de contrato.
13. Iterar sobre otro artefacto.
14. Modelo en incrementos pequeños.
15. Modelar para comunicar.
16. Modelar para comprender.
17. Modelar con otros.
18. Poner a prueba con código.
19. Reutilizar los recursos existentes.
20. Actualizar sólo cuando duele.
21. Utilizar las herramientas más simples (CASE, o mejor pizarras, tarjetas, post-its).

Como AM se debe usar como complemento de otras metodologías, nada se especifica sobre métodos de desarrollo, tamaño del equipo, roles, duración de iteraciones, trabajo distribuido y criticidad, todo lo cual dependerá del método que se utilice.

Los diagramas de UML y los artefactos del Proceso Unificado, por ejemplo, han sido explorados en extremo detalle describiendo cómo debería ser su tratamiento en un proceso ágil EUP. EUP es simplemente UP+AM.

3.3.2.10. **Pragmatic Programming (PP)**

El título en realidad no es totalmente cierto, ya que no hay un método llamado programación pragmática, sino que existe un conjunto muy interesante de las mejores prácticas de programación publicadas en el libro “The Pragmatic Programmer” [23], pero por practicidad se lo llama Programación Pragmática [24].

La programación pragmática no tiene proceso, fases, roles distintos o productos de trabajo. Esta metodología ágil cubre, sin embargo, la mayoría de las mejores prácticas de programación. Hay un total de 70 recomendaciones que se enfocan en los problemas del día a día, pero la filosofía por detrás de esto es que son universales y pueden ser aplicadas a cualquier fase del desarrollo de software.

La filosofía está definida en 6 puntos:

1. Tome las responsabilidades para Ud., piense en soluciones en lugar de estar pensando en excusas.
2. No diseñe o codifique mal, arregle las inconsistencias o planee arreglarlas tan pronto como sea posible.
3. Tome un rol activo para introducir cambios donde Ud. vea que son necesarios.
4. Haga software que satisfaga a su cliente, pero sepa cuándo parar.
5. Constantemente amplíe su conocimiento.
6. Mejore sus habilidades de comunicación.

Desde un punto de vista ágil, la mayoría de las prácticas más interesantes se enfocan sobre el desarrollo iterativo, incremental, testing riguroso y diseño centrado en el usuario. El enfoque tiene un punto de vista muy práctico, y la mayoría de las prácticas son ilustradas con ejemplos positivos y negativos, a menudo complementados con preguntas y trocitos de código. Es un esfuerzo considerable explicar cómo diseñar e implementar software tal que pueda resistir cambios. Una de las soluciones que se plantean para mantener software a través de los cambios es la refactorización.

El enfoque de la Programación Pragmática respecto del testing consiste en testear el código que está siendo implementado sobre el código real, y todas las pruebas deben ser hechos de forma automática. La idea es que si cada fallo corregido no es agregado dentro de la biblioteca del test y si los tests de

regresión no se corren periódicamente, el tiempo y el esfuerzo se gastan en encontrar los mismos fallos repetidamente, y los efectos adversos de cambios en el código no pueden detectarse suficientemente temprano.

La automatización se encuentra en la PP en algunas otras tareas también. De hecho llega a sugerir “No use procedimientos manuales”. Por ejemplo, en la creación de la primera documentación de código fuente y en la creación de código de las definiciones de la base de datos.

Recomienda conservar las especificaciones a un nivel razonable de detalle. La PP demuestra prácticas de software simples, responsables y disciplinadas. La prácticas sugeridas en PP son escritas desde el punto de vista de un programador, independientemente de los métodos o procesos que se utilicen, para evitar errores típicos en la codificación y en el diseño y errores de comunicación en el grupo de trabajo.

3.4. Clásicas Vs Ágiles:

Los métodos ágiles cambian significativamente algunos de los énfasis de las Metodologías “clásicas”: [25]

- Las metodologías ágiles, son metodologías adaptables en lugar de predictivas. Las metodologías clásicas tienden a intentar planear una parte grande del desarrollo del software en gran detalle para un plazo grande de tiempo. Esto funciona bien, siempre y cuando no se produzcan cambios en el proyecto, cosa que la experiencia demuestra que en un gran número de proyectos se producen. Por esto podemos destacar las metodologías ágiles como muy resistentes a los cambios que se producen a lo largo de la vida de un proyecto, siendo aceptable hasta el cambio de metodología en el desarrollo del proyecto.
- Las metodologías ágiles están orientados a las personas y no orientados al proceso. El fin de las metodologías clásicas es definir un proceso que puede ser utilizado por cualquiera que lo use. Las metodologías ágiles apuestan por que la metodología no limite las habilidades del equipo de desarrollo, por tener que estar ceñidos a un proceso. Las metodologías ágiles apuestan por trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

3.4.1. Separación de Diseño y Construcción:

La inspiración usual para las metodologías han sido las ingenierías civil o mecánica. Estas disciplinas resaltan que hay que planear antes de construir. Los ingenieros trabajan sobre una serie de esquemas que indican precisamente qué hay que construir y como se deben construir. En estas ingenierías se desarrolla un diseño y una planificación, los cuales suelen ser entregados a otras personas o empresas que se encargan de la construcción, en función del diseño y planificación entregada. Este plan define las tareas que necesitan hacerse y las dependencias que existen entre estas tareas. También se detalla cómo deben hacer su trabajo las personas que participan en la

construcción. Esto permite que la construcción requiera menos pericia intelectual, aunque se necesite mucha habilidad manual.

Por lo planteado en el párrafo anterior podemos ver como en dichas ingenierías existen dos actividades bastante diferenciadas. El diseño, qué es difícil de predecir y requiere personal especializado y costoso, y que suele suponer un 10% de tiempo del proyecto, y la construcción que requiere un equipo menos especializado, es más fácil de predecir, y supone un 90% de tiempo del proyecto.

Tras asumir este planteamiento nos pueden surgir las siguientes preguntas. ¿Es posible armar un plan que sea capaz de convertir el código en una actividad de construcción predecible? Y en tal caso, ¿es la construcción suficientemente grande en costo y tiempo para hacer valer la pena este enfoque?

Todo esto trae a la mente más preguntas. La primera es la cuestión es como de difícil es conseguir un diseño UML en un estado que pueda entregarse a los programadores. El problema con un diseño tipo UML es que puede parecer muy bueno en el papel, pero resultar seriamente fallido a la hora de la programación. La única verificación que podemos hacer con los diagramas UML es la revisión cuidadosa. Mientras esto es útil trae errores al diseño que sólo se descubren durante la codificación y pruebas.

Por otra parte en software la cantidad de tiempo que se invierte en construcción (codificar) es mucho menor. Normalmente en un proyecto grande, sólo 15% del proyecto son código y pruebas unitarias. Aun cuando se consideren las pruebas parte de la construcción, el plan supondría el 50% del total.

Esta clase de preguntas llevaron a Jack Reeves a sugerir [26] que el código fuente es un documento de diseño y que la fase de construcción está en realidad en la compilación y el ligado. Esta idea lleva a las siguientes conclusiones:

- En software la construcción es tan barata que es casi gratis.

- En software todo el esfuerzo está en el diseño, de modo que requiere de personas creativas y talentosas.
- Los procesos creativos no se planean fácilmente, de modo que la previsibilidad bien puede ser una meta imposible.
- Debemos ser muy cautos al usar la metáfora de la ingeniería tradicional para construir software. Es un tipo diferente de actividad y por tanto requiere un proceso diferente.

3.4.2. La imprevisibilidad de los requisitos:

En el negocio de construcción de software los cambios en los requisitos son la norma, la pregunta es qué se puede hacer al respecto.

La idea detrás de la ingeniería de requisitos es conseguir un esquema totalmente entendido de los requisitos antes de empezar a construir el software y conseguir la firma del cliente sobre estos requisitos.

En un ambiente de construcción de software es muy difícil tener dicho esquema ya que no es una tarea fácil entender las necesidades del cliente, ya que el cliente no sabe expresarlas en términos de software, y es complejo entender en profundidad su negocio o actividad.

De primera mano el valor que proporciona un software es difícil de cuantificar, solo cuando se comienza a usar es posible determinar la implementación de que requisitos aporta más valor. Muchos cambios en el mundo de negocios son completamente imprevisibles y afectarán enormemente a los requisitos.

Aun si dispusiéramos de unos requisitos inamovibles, calcular el costo de implementación de esos requisitos puede ser complejo por varias razones:

- Porque los materiales básicos cambian rápidamente. Las tecnologías evolucionan rápidamente y en ocasiones son complejas de utilizar.
- Por lo mucho que depende el software de los individuos involucrados, y los individuos son difíciles de predecir y cuantificar.

Casi todo en el desarrollo de software depende de los requisitos. Si no se pueden obtener requisitos estables no se puede obtener un plan predecible.

3.4.3. ¿Es imposible la previsibilidad?

Todo esto nos lleva a la conclusión de que la actividad de desarrollar software es imposible o muy difícil de predecir o planear. Como en tantos problemas la parte más difícil está simplemente en comprender que el problema existe.

Por tanto el uso de metodologías predictivas no parece lo más adecuado, para la gran mayoría, de los desarrollos de software.

Las metodologías ágiles tratan de buscar respuesta a este problema siendo conscientes de que existe la previsibilidad, lo que no significa que haya que volver a un caos ingobernable. Más bien hace falta un proceso que pueda dar control sobre la imprevisibilidad. De eso se trata la adaptabilidad.

La adaptabilidad se fundamenta en dos pilares fundamentales:

- Construcción iterativa del software.
- Relación más estrecha con el cliente.

3.4.4. Controlando un proceso imprevisible. Las Iteraciones

Para controlar los procesos imprevisibles necesitamos un mecanismo de retroalimentación que pueda decirnos con precisión cuál es la situación a intervalos frecuentes. Dicho mecanismo es el desarrollo iterativo. La clave del desarrollo iterativo es producir frecuentemente versiones funcionales del sistema final que tengan un subconjunto de los rasgos requeridos. Estos sistemas son cortos en funcionalidad, pero por otra parte deben ser fieles a las demandas del sistema final. Deben estar totalmente integrados y tan cuidadosamente probados como una entrega final.

El desarrollo iterativo también tiene sentido en los procesos predictivos. Pero es esencial en los procesos adaptables, ya que un proceso adaptable necesita poder tratar con los cambios. Esto lleva a un estilo de planear, donde los planes a largo plazo son muy fluidos, y los únicos planes estables son a corto plazo hechos para una sola iteración.

3.4.5. Orientado a la gente

Uno de los objetivos de las metodologías tradicionales es desarrollar un proceso donde las personas involucradas sean partes reemplazables. Con tal proceso se puede tratar a las personas como recursos que están disponibles en varios tipos. Se tienen un analista, algunos programadores, algunos testadores, un gerente. Los individuos no son tan importantes, sólo los papeles lo son.

Pero realmente ¿son las personas involucradas en el desarrollo de software partes reemplazables? Uno de los rasgos importantes de los métodos ágiles es el rechazo a esta afirmación.

La creación de software es un proceso creativo y que requiere talento, ni es fácil medir el talento de las personas ni todas las personas tienen el mismo talento.

La noción de la gente como recursos está profundamente inculcada en el pensamiento de negocios, teniendo sus raíces en el impacto del enfoque de La Dirección Científica de Frederick Taylor.

Una parte clave de la noción Taylorista es que la gente que hace el trabajo, no es la mejor gente para entender la mejor manera de hacer el trabajo.

Cuando se quiere contratar y retener a gente capaz, hay que reconocer que son profesionales competentes. Como tales, son los mejores para decidir cómo dirigir su trabajo técnico.

Los desarrolladores deben poder tomar todas las decisiones técnicas. Sólo los desarrolladores pueden estimar cuánto tiempo se va a emplear en hacer un trabajo.

Por la naturaleza creativa del desarrollo de software las diferencias entre los buenos y malos desarrolladores son enormes, retener a los buenos programadores y proporcionándoles un entorno de trabajo adecuado se hace indispensable.

3.4.6. El Cliente Adaptable

Este tipo de proceso adaptable requiere un tipo diferente de relación con el cliente a la realizada en los procesos fijos, sobre todo cuando el desarrollo es hecho por otra empresa. La mayoría de los clientes prefieren un contrato a precio fijo. Un contrato a precio fijo requiere requisitos estables y por tanto procesos predictivos. Los procesos adaptables y los requisitos inestables implican que no se puede trabajar con la noción usual de precio fijo. Tratar de encajar un modelo de precio fijo a un proceso adaptable puede acabar de manera desastrosa. Este desastre provoca que tanto el cliente como la empresa desarrolladora de software queden en muy mal lugar. Después de todo, el cliente no querría un software a menos que su negocio lo necesitara. Si no lo consigue su negocio sufre. Así aun cuando no pague nada a la compañía de desarrollo, todavía pierde. De hecho pierde más de lo que pagaría por el software.

De modo que hay peligro para ambos lados al firmar un contrato a precio fijo en condiciones donde un proceso predictivo no puede usarse. Esto significa que el cliente tiene que trabajar de otro modo.

Esto no significa que no se pueda fijar un presupuesto para software por adelantado. Lo que significa es que no se puede fijar el tiempo, precio y alcance. La manera ágil usual es fijar tiempo y precio y permitir que el alcance varíe de manera controlada.

En un proceso adaptable el cliente tiene mucho control sobre el proceso de desarrollo de software. A cada iteración puede tanto verificar el progreso como alterar la dirección del desarrollo de software. Esto lleva a una relación mucho más íntima con los desarrolladores de software, una verdadera sociedad de trabajo. Este nivel de compromiso no es para cualquier organización o cliente, ni para cualquier desarrollador de software; pero es esencial para lograr que un proceso adaptable funcione apropiadamente.

El beneficio importante para el cliente es un desarrollo de software mucho más sensible. Un sistema usable, aunque mínimo, puede entrar en producción pronto. El cliente puede cambiar sus capacidades de acuerdo a los cambios en el negocio, y también aprender cómo se usa el sistema en realidad.

Una pieza tan importante como ésta es una visibilidad mayor sobre el verdadero estado del proyecto. El problema con los procesos predictivos es que la calidad del proyecto se mide por la conformidad con el plan. Esto dificulta a la gente señalar cuándo la realidad y el plan divergen. El resultado común es un gran resbalón más tarde en el calendario del proyecto. En un proyecto ágil hay un constante rehacer del plan con cada iteración. Si las malas noticias están al acecho, tienden a aparecer más temprano, cuando aún se puede hacer algo al respecto. De hecho este control del riesgo es una ventaja clave del desarrollo iterativo. Los métodos ágiles van más allá, manteniendo corta la duración de la iteración, pero también viendo estas variaciones como oportunidades.

Hay un aspecto importante en lo que constituye un proyecto exitoso. Un proyecto predictivo se mide a menudo por lo bien que coincide con el plan. Un proyecto que está a tiempo y en costo es un éxito. Esta medida no tiene sentido en un ambiente ágil. Para el desarrollo ágil la cuestión es el valor de negocio (beneficio) que el cliente consiga, es decir, un software más valioso que el costo que puso en él. Un buen proyecto predictivo irá de acuerdo al plan, un buen proyecto ágil construirá algo diferente y mejor que lo que se esperaba en el plan original.

3.4.7. Costo de los Cambios en la Construcción de SW

En la ilustración 17 se muestra el costo de producir cambios en el software que se desarrolla mediante una metodología tradicional frente el costo de producir cambios en el software que se desarrolla mediante alguna de las metodologías ágiles. Como se puede apreciar, a medida que avanza el tiempo, el costo es exponencial en el caso de la construcción mediante una metodología tradicional.

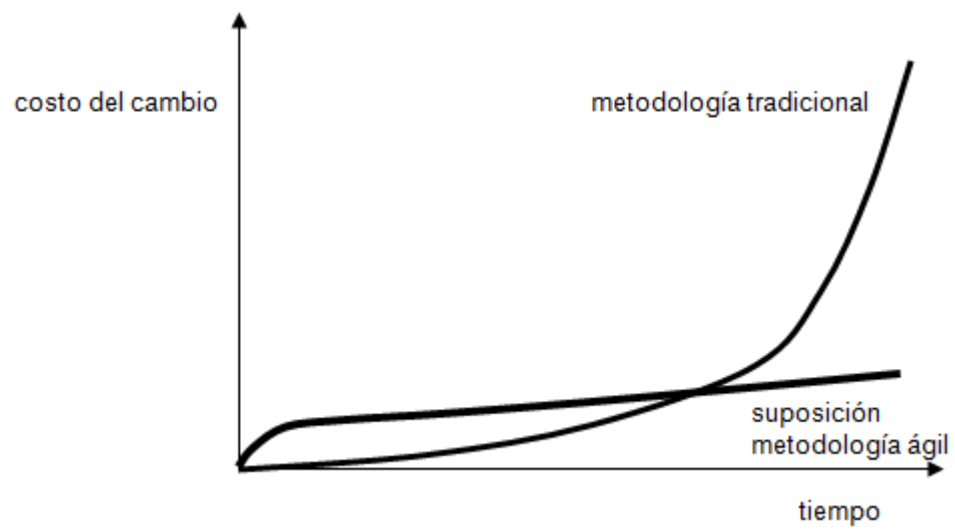


Ilustración 17. Función de costos de cambio en el software desarrollado

3.4.8. Comparación Ágil - Tradicional

En la tabla 1 podemos ver comparada, de forma resumida, los puntos de divergencia que se producen a la hora de enfrentar las metodologías ágiles contra las metodologías tradicionales.

Metodología Ágil	Metodología Tradicional
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Especialmente preparados para cambios durante el Proyecto.	Cierta resistencia a los cambios.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Grupos grandes.
Menos énfasis en la arquitectura.	La arquitectura es esencial.

Tabla 1. Metodología tradicional contra metodología ágil

4 **Gestión Multiproyecto:**

4.1. Definiciones y objetivos

Cuando se trabaja en un ambiente multiproyecto, las decisiones que se tomen en un proyecto concreto afectan a otros proyectos aunque no se encuentren en ejecución o la relación entre los proyectos no sea evidente. [27]

Como ya hemos visto todos los proyectos, incluso los mejor planificados, tienen un alto grado de incertidumbre. Esto es debido al gran número de variables involucradas como pueda ser: la duración de las actividades, su complejidad, y la habilidad en la ejecución.

Los proyectos en un portfolio interfieren entre sí; ya que todos los proyectos están directa o indirectamente relacionados, como por ejemplo, por el simple hecho de compartir recursos, ya sean materiales o humanos.

El ambiente multiproyecto, puede considerarse como un sistema con comportamiento complejo, debido al gran número de interacciones, la respuesta es no lineal y dependiente del tiempo.

4.2. El rol de la Oficina de Proyectos

La oficina de proyectos (PMO) es la entidad organizadora encargada de gestionar el trabajo multiproyecto. Su objetivo principal, por contraposición a la de un proyecto en particular, es el de finalizar todos los proyectos de manera que se vean satisfechos los objetivos de la organización [27].

La PMO puede actuar como fuente de información de mejores prácticas de gestión de proyectos y/o ser la responsable de los programas y proyectos de la organización.

El Gartner Group define tres posibles roles para la PMO:

1. **PMO tipo repositorio:** Es el rol más básico de PMO y se encarga de los procesos para gestionar proyectos y archivo de mejores prácticas. En este caso la PMO no se involucra formalmente en decisiones correspondientes a la ejecución de los proyectos, siendo responsabilidad de las áreas funcionales correspondientes.

2. **PMO tipo consultor:** En este tipo la PMO adopta un rol más activo. Provee de una guía a los gestores de los proyectos y participa en las revisiones de los proyectos. También ayuda en el comienzo y cierre de cada proyecto y provee experiencia en áreas especializadas, como por ejemplo, gestión del riesgo y estimación de costos. Ejecuta algunas tareas de monitoreo y control de los proyectos, pero no ordena ninguna acción correctiva.
3. **PMO tipo gerencial:** Esta PMO actúa como un agente gerencial. A parte de asumir las responsabilidades de los dos anteriores, es responsable de preparar el plan maestro y de recursos reflejando el trabajo que el área de IT se ha comprometido a efectuar en el corto plazo y también prevé los objetivos de los proyectos y monitorea la ejecución del portfolio.

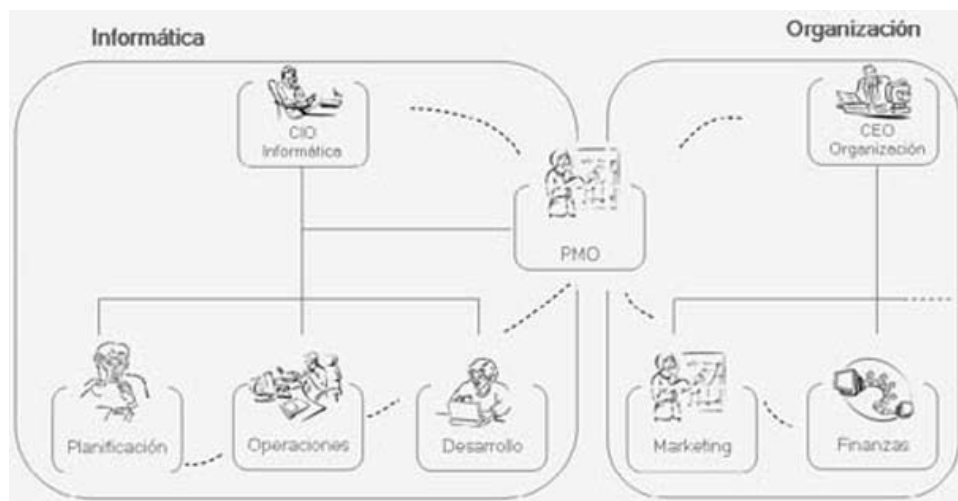


Ilustración 18. PMO tipo gerencial

En función del tipo de PMO que se disponga, ésta deberá administrar un número mayor o menor de procesos y por tanto tendrá que variar la forma de adaptarlos. También en función del rol de la PMO que se tenga, puede que alguno de los procesos no los realice la propia PMO sino que sean llevados a cabo por alguna otra área.

4.3. Procesos de la Oficina de Proyectos

Un ambiente multiproyecto requiere que se definan procesos comunes, ya que estos proporcionan un lenguaje común, facilitando la comunicación entre proyectos y disciplinas, y la incorporación y formación de personal nuevo. La definición de procesos comunes también facilita poder capturar el conocimiento colectivo desarrollado a través de la experiencia del personal, mejorando los procesos y las herramientas utilizadas.

Existen tres grandes grupos de procesos llevados adelante por la PMO. Estos son:

1. Gestión del ciclo de vida de los proyectos:

El curso de un proyecto desde el punto vista del PMO pasa por los procesos de:

- 1.1. Formulación del Proyecto: Cuando se recibe una solicitud de un nuevo proyecto o una modificación considerable de un proyecto ya existente, la PMO realiza un estudio preliminar del alcance, enfoque metodológico, duración, y esfuerzo requerido. Toda esta información permite a la PMO tomar la decisión de si el proyecto será introducido en el portafolio o no.
- 1.2. Inicio del Proyecto: Se forma el equipo principal del proyecto y se revisan los aspectos estudiados en el punto anterior, mostrando especial hincapié en la disponibilidad de recursos, acordando los cambios necesarios con el cliente. Los cambios que pudieran llegar a afectar en otros proyectos son analizados para su aprobación, teniendo en cuenta el plan maestro de proyectos. A partir de este momento se definen los paquetes de trabajo, y se crean los centros de costo. Se establece una referencia contra la cual se medirá el rendimiento. Por último se asegura la disponibilidad de recursos y del trabajo de acuerdo al plan del proyecto.
- 1.3. Ejecución del Proyecto: El líder del proyecto debe producir los resultados deseados, fomentar el trabajo en equipo y el compromiso, y asegurar que se siguen los procesos, metodologías y estándares de la organización.

- 1.4. Cierre del Proyecto: Cuando el proyecto finaliza, la PMO debe asegurarse de que todo el trabajo ha finalizado, de que se incorpore la experiencia generada como conocimiento que sirva para futuros proyectos, y cerrar los contratos con subcontratas y proveedores.
- 1.5. Auditoría del Proyecto: Una auditoría nos proporciona el estado real de un proyecto, y debe ser llevado por una persona que no pertenezca al equipo del proyecto. Los objetivos son:
 - 1.5.1. Certificar que el trabajo se está desarrollando de acuerdo a los procedimientos definidos.
 - 1.5.2. Fijar el estado final del proyecto, en términos de tiempo, calidad, costo, alcance, satisfacción del cliente y moral de los empleados.

2. Administración del Portfolio de Proyectos:

- 2.1. Planificación del Portfolio de Proyectos: En este proceso se concretan las necesidades del negocio para decidir la cartera de proyectos. El resultado de este proceso es un plan que balancea trabajo, resultados, recursos y riesgo de acuerdo a los objetivos de la organización. Al finalizar este proceso se conocerá qué proyectos serán asumidos y cuándo se desarrollaran, prediciendo los recursos necesarios para ejecutar los seleccionados, y proyectando los flujos de caja correspondientes.
- 2.2. Revisión de los Proyectos: El propósito de esta fase es generar alertas tempranas acerca del rendimiento de proyectos individuales, de tal manera que la gerencia pueda actuar antes de que un problema local afecte a toda la cartera de proyectos. Dichas alertas se generan comparando el rendimiento real del proyecto contra una referencia preestablecida, y de ser posible contra datos históricos de proyectos similares. Las áreas a analizar serán progreso, costo, calidad.
- 2.3. Control del Portfolio: Es el proceso mediante el cual la PMO lleva a cabo un proceso para minimizar el impacto de los desvíos en un proyecto concreto sobre toda la cartera. Las acciones correctivas que haya que tomar se deciden atendiendo al marco del plan maestro de recursos y proyectos, y no considerando exclusivamente el proyecto afectado.

3. Tareas de Apoyo:

- 3.1. Gestión Procesos y Sistemas de Información: La importancia de una buena gestión por parte de los gestores, se fundamenta en la creación y utilización de procesos y herramientas comunes. Es muy importante que exista el concepto de mejora del proceso en la PMO, para ello se debe detectar los cuellos de botella de los procesos (actividades limítrofes según algún indicador tanto en calidad como costos) y atacarlos. Una vez solucionado dicho cuello de botella, aparecerá otro en otra parte del proceso y habrá que repetir la tarea.
- 3.2. Mediciones e Indicadores: Esta tarea tiene tres actividades: Planificar los indicadores, llevar adelante las mediciones y generar estadísticas.
- 3.3. Administración de Cambios: Los cambios en el proyecto son naturales, y parte del trabajo del proyecto consiste en dar respuesta a cambios en el ambiente y los deseos o necesidades de los clientes. Administración de cambios es un proceso que está vinculado a todo el trabajo que se efectúa en el proyecto. Se tiene que tener en cuenta los cambios en el alcance, y también la evolución de los productos generados durante el desarrollo. Este proceso se desarrolla a lo largo de tres tareas: Gestión de requerimientos, gestión de la configuración y configuración.
- 3.4. Adquisiciones: El fin de esta actividad es ofrecer apoyo a los gerentes de proyecto para tratar con terceras partes, vendedores y subcontratistas involucrados en sus proyectos.
- 3.5. Aseguramiento de la Calidad: El proceso de aseguramiento de la calidad es el consistente en el control periódico de los productos finalizados y los procesos de trabajo elaborados. Tareas que se realizan en este proceso son:
 - 3.5.1. Confronta los productos con las especificaciones y evalúa el nivel de alcance.
 - 3.5.2. Comprueba que el trabajo se haga de acuerdo a procesos y estándares.
 - 3.5.3. Proporciona información al personal del proyecto, sobre los resultados de las actividades de Calidad.
 - 3.5.4. Hace un seguimiento de los problemas encontrados y se asegura de que sean resueltos.

- 3.6. Soporte Administrativo: Este proceso es el encargado de la gestión administrativa interna de la oficina de proyectos.
- 3.7. Contabilidad del Proyecto: Este proceso es el encargado de analizar, almacenar y generar reportes sobre todos los eventos financieros del proyecto.
- 3.8. Gestión de Recursos Humanos: La PMO debe seleccionar, contratar, y formar personal con habilidades de gestión de proyectos y soporte de proyectos. Para ello, debe preparar programas de entrenamiento y capacitaciones, y trabajar en conjunto con los recursos humanos para establecer planes de carrera y mecanismos de recompensa adecuados.

5 Tecnologías:

5.1. VMware:

VMware es un sistema de virtualización por software, desarrollado por VMWare Inc. Es una aplicación que simula un sistema físico (un computador, un hardware) con unas características de hardware determinadas. Cuando se ejecuta la aplicación (simulador), proporciona un ambiente de ejecución similar, a todos los efectos, a un computador físico (excepto en el puro acceso físico al hardware simulado), con CPU, BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc, pudiendo configurar cada uno de ellos y permitiendo añadir el número deseado (en función del dispositivo puede estar limitado por el número de dispositivos físicos de la máquina que ejecuta VMware).

VMware permite simular varios sistemas operativos dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos suficiente para usarse en entornos de producción.

Existen varias versiones gratuitas, como son VMware vSphere Hypervisor VMware Server y VMware Player, y gran multitud de versiones de pago, destacando VMware Workstation, siendo esta la utilizada para desarrollar las pruebas realizadas.

5.2. OpenSuse:

OpenSUSE es el nombre de la distribución y proyecto libre creado por Novell y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux. Después de adquirir SUSE Linux en enero de 2004, Novell decidió lanzar SUSE Linux Professional como un proyecto completo de código abierto, involucrando a la comunidad en el proceso de desarrollo.

Versión 11.3:

Esta versión de OpenSuse incluye numerosas aplicaciones de escritorio, aplicaciones de servidor (WWW, mail, FTP,...) y una selección de herramientas de desarrollo.

Otro de los aspectos en los que han trabajado duro en openSUSE 11.3 ha sido la compatibilidad con los principales dispositivos móviles actuales. Esta nueva versión permite sincronizar música, acceder a ficheros gráficos o compartir la conexión a internet con dispositivos Android, iPhone de Apple o Blackberry.

Zypper, el gestor de paquetes por línea de comandos, ha mejorado la gestión de dependencias, encontrado su alternativa grafica en el gestor Yast2.

La base del sistema dispone del Kernel 2.6.34, la versión 1.0.23 de los drivers ALSA, X.org 7.5, KMS activado para tarjetas ATI, Intel y Nvidia, drivers Nouveau y Radeon, Upstart como sistema de arranque, la posibilidad de probar Grub2 y el sistema contrack para el filtrado del tráfico de la red.

Los entornos de escritorio disponibles son GNOME 2.30.1, con la posibilidad de probar la versión 3.0; KDE SC 4.4.4 y LXDE Desktop, además de incluir la posibilidad de instalar Xfce 4.6.1. Entre las principales aplicaciones disponibles: GoogleCL, OpenOffice.org 3.2.1, Mozilla Firefox 3.6.6, Rosegarden 10.04 o la versión 3.0.5 de Thunderbird.

Entre las herramientas de desarrollo incluidas destacan los compiladores GCC 4.5, GDB 7.1 o Mono 2.6.4, los entornos Netbeans 6.8, KDevelop 4.0, Qt-Creator 1.3.1 y Monodevelop 2.2.2. OpenSUSE 11.3 incluye como novedad los paquetes de los servidores MariaDB y MySQL Cluster.

Por lo demás, OpenSUSE mantiene su estabilidad y robustez a todos los niveles, junto con un amplio catálogo de software, que la convierten en una de las mejores distribuciones GNU/Linux disponibles.

5.3. *Windows XP:*

Windows XP (cuyo nombre en clave inicial fue *Whistler*) es un sistema operativo desarrollado por Microsoft Windows. Lanzado al mercado el 25 de octubre de 2001, actualmente es el segundo sistema operativo para x86 más utilizado del mundo (con una cuota de mercado del 20,58%⁴) y se considera que existen más de 400 millones de copias funcionando. Las letras "XP" provienen de la palabra *eXPerience*.

Dispone de versiones para varios entornos informáticos, incluyendo PCs domésticos o de negocios, además de equipos portátiles, netbooks, tabletPCs y media center. Sucesor de Windows 2000 junto con Windows ME, y antecesor de Windows Vista, es el primer sistema operativo de Microsoft orientado al consumidor que se construye con un núcleo y arquitectura de Windows NT disponible en versiones para plataformas de 32 y 64 bits.

A diferencia de versiones anteriores de Windows, al estar basado en la arquitectura de Windows NT, presenta mejoras en la estabilidad y el rendimiento. Tiene una interfaz gráfica de usuario (GUI) perceptiblemente reajustada (denominada *Luna*), la cual incluye características rediseñadas, algunas de las cuales se asemejan ligeramente a otras GUI de otros sistemas operativos, cambio promovido para un uso más fácil que en las versiones anteriores. Se introdujeron nuevas capacidades de gestión de software para evitar el *infierno de las DLLs* que plagó las viejas versiones. Ha sido también criticado por las vulnerabilidades de seguridad, integración de Internet Explorer, la inclusión del reproductor Windows Media Player y aspectos de su interfaz.

5.4. *Windows 2003 Server:*

Windows Server 2003 es un sistema operativo de la familia Windows de la marca Microsoft para servidores que salió al mercado en el año 2003. Está basada en tecnología NT y su versión del núcleo NT es la 5.2.

En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado para labores empresariales, no con menos funciones,

⁴ Dato obtenido el 12 de Agosto de 2013 en <http://gs.statcounter.com/>

sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las características de servidor.

Sus características más importantes son:

- Sistema de archivos NTFS: con cuotas, cifrado y compresión de archivos, carpetas y no unidades completas. También permite montar dispositivos de almacenamiento sobre sistemas de archivos de otros dispositivos al estilo unix
- Gestión de almacenamiento, backups... incluye gestión jerárquica del almacenamiento, consistente en utilizar un algoritmo de caché para pasar los datos menos usados de discos duros a medios ópticos o similares más lentos, y volverlos a leer a disco duro cuando sean necesarios.
- Windows Driver Model: Implementación básica de los dispositivos más utilizados, de esa manera los fabricantes de dispositivos sólo han de programar ciertas especificaciones de su hardware.
- ActiveDirectory Directorio de organización basado en LDAP, permite gestionar de forma centralizada la seguridad de una red corporativa a nivel local.
- Autenticación Kerberos5.
- DNS con registro de IPs dinámicamente.
- Políticas de seguridad.

Los servidores que maneja Windows 2003 son:

- Servidor de archivos.
- Servidor de impresiones.
- Servidor de aplicaciones.
- Servidor de correo (SMTP/POP).
- Servidor de terminal.
- Servidor de Redes privadas virtuales (VPN) (o acceso remoto al servidor).
- Controlador de Dominios (mediante Active Directory).
- Servidor DNS.

- Servidor DHCP.
- Servidor de Streaming de Vídeo.
- Servidor WINS.
- Servidor RIS Remote Installation Services (Servicios de instalación remota).

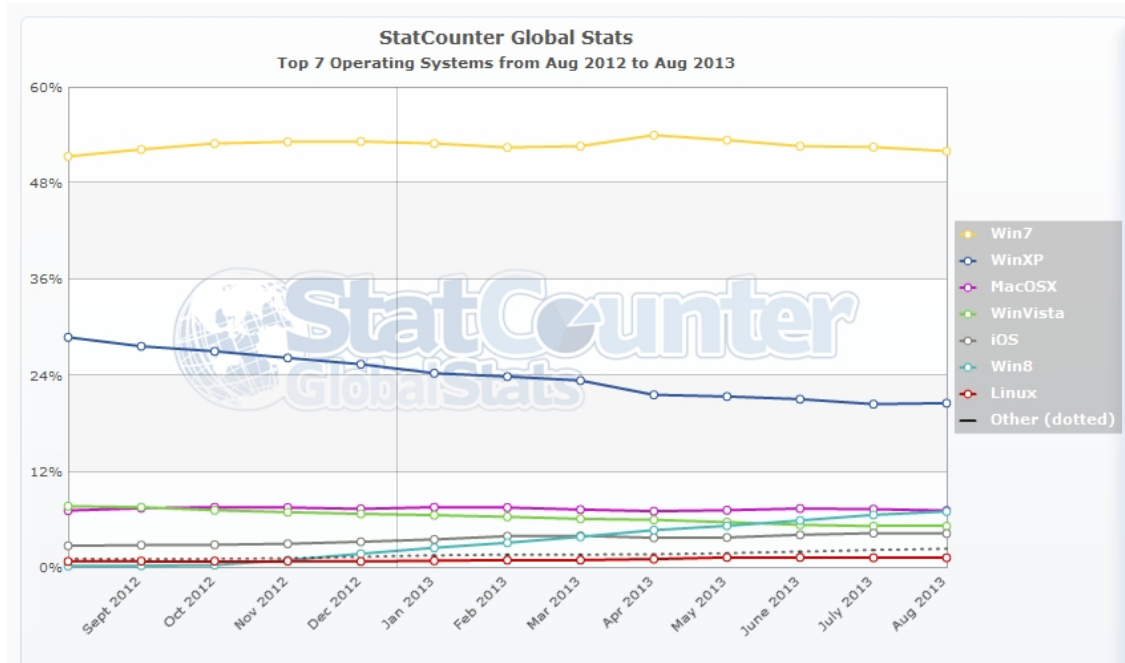


Ilustración 19. Cuota de utilización mundial de sistemas operativos.

5.5. MySQL:

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario que otros proyectos de software libre, donde es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

5.6. Apache:

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado (47% de cuota⁵). Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de previsualizar y probar código mientras éste es desarrollado.

Microsoft Internet Information Services (IIS) es el principal competidor de Apache. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS).

⁵ Dato obtenido el 10 de Agosto de 2013 en <http://news.netcraft.com/archives/2013/08/09/august-2013-web-server-survey.html>

5.7. IIS:

Internet Information Services o **IIS** es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Los servicios de Internet Information Services proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

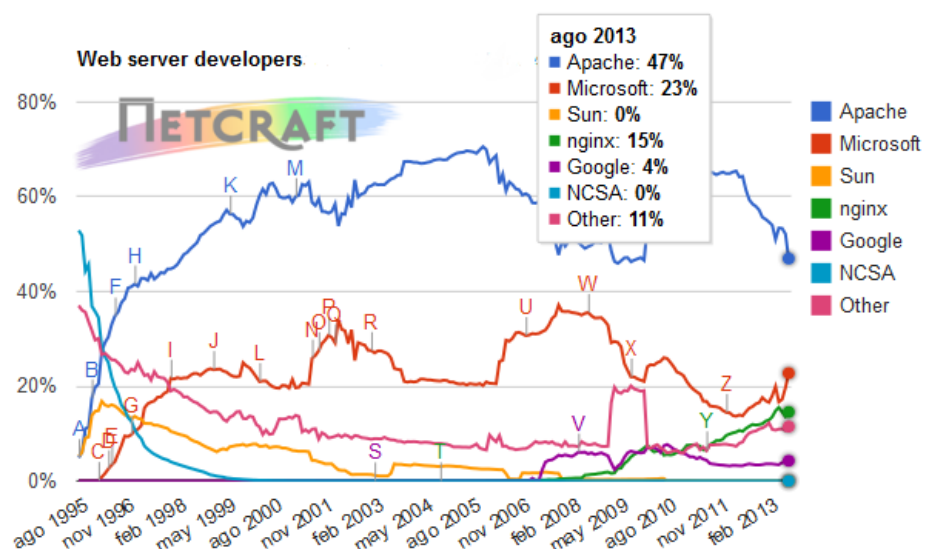


Ilustración 20. Porcentaje de servidores web usados mundialmente.

5.8. ASP.NET:

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Cualquier persona que está familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de *script* y plataformas de servidor. Por desgracia para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones web tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.

Microsoft introdujo la tecnología llamada Active Server Pages en diciembre de 1996. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. La definición contextual de Microsoft es que *"Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web"*.

Después del lanzamiento del Internet Information Services 4.0 en 1997, Microsoft comenzó a investigar las posibilidades para un nuevo modelo de aplicaciones web que pudiera resolver las quejas comunes sobre ASP, especialmente aquellas con respecto a la separación de la presentación y el contenido y ser capaz de escribir código "limpio".

La primera demostración pública y la liberación de la primera beta de ASP+ (y el resto del .NET Framework) se realizó en el Microsoft's Professional Developers Conference (PDC) el 11 de julio de 2000 en Orlando, Florida. Durante la presentación de Bill Gates, Fujitsu demostró ASP+ usado en conjunción con COBOL, y el soporte para una variedad de otros lenguajes fue anunciada, incluyendo los nuevos lenguajes de Microsoft, Visual Basic .NET y C#, así como también el soporte por medio de herramientas de interoperabilidad para Python y Perl creadas por la empresa canadiense ActiveState.

Una vez que la marca ".NET" fue seleccionada en la segunda mitad del 2000, se cambió el nombre de ASP+ a ASP.NET. Mark Anders explicó en una aparición en *The MSDN Show* en ese año.

5.9. Java:

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

5.10. PHP:

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio.

6 Tipos de aplicaciones de gestión de proyectos

De las aplicaciones evaluadas se pueden hacer dos grupos diferenciados atendiendo a la forma de trabajar con ellas.

6.1. Aplicaciones de Escritorio:

Son aquellas aplicaciones que se instalan y ejecutan en un ordenador personal sobre un sistemas operativo (Windows, Linux, MacOS...). Los datos de la aplicación son almacenados en su disco duro y el rendimiento de la aplicación depende de la configuración hardware (procesador, memoria ram, de video, etc.) que posea la máquina sobre la que se trabaja.

- Ventajas:
 - Poseen un menor tiempo de respuesta.
 - Pueden ser más robustas.
 - Mayor capacidad gráfica visual.
 - Mayor personalización
- Desventajas:
 - Requiere instalación.
 - Son desarrolladas para un sistema operativo concreto.
 - En caso de actualización, se debe de realizar en cada cliente.

6.2. Aplicaciones Web:

Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web, situado en internet o en una intranet, utilizando un navegador web. Los datos de la aplicación se almacenan en un servidor web.

- Ventajas:
 - Se puede usar desde cualquier lugar en el que se disponga de un navegador.
 - No ocupan espacio en el disco duro.
 - No requieren hacer actualizaciones en los clientes.
 - No hay problemas de incompatibilidad entre versiones, porque todos los clientes trabajan con la misma versión.
 - No se obliga a usar un sistema operativo concreto.

- Desventajas:
 - Requiere conexión a la red para poder acceder a la aplicación.
 - Suelen ofrecer menos funcionalidades que las aplicaciones de escritorio.
 - Se pierde tiempo de desarrollo haciendo la aplicación compatible con los distintos navegadores.
 - Su tiempo de respuesta es más lento.

Atendiendo a la seguridad, cualquiera de los dos tipos de aplicaciones se puede hacer de manera segura o insegura, ya que depende de los desarrolladores, pero las aplicaciones web, al estar alojadas en un servidor dedicado, suelen contar con más medidas de seguridad que las aplicaciones de escritorio alojadas en ordenadores personales.

7 **Aplicaciones evaluadas:**

7.1. OpenProject:

Es una aplicación de escritorio para la gestión de proyectos, de libre distribución y código OpenSource, desarrollado a partir de varias versiones betas. Actualmente se encuentra en la versión 1.4 bajo licencia CPAL 1.0 (Common Public Attribution License).

Es una aplicación multiplataforma, ya que está desarrollada en Java, por lo que permite ser instalado y ejecutado en máquinas Windows, Linux y MacOSX.

Permite realizar gráficas de Gantt⁶ y diagramas de PERT⁷ de las tareas que componen el proyecto.

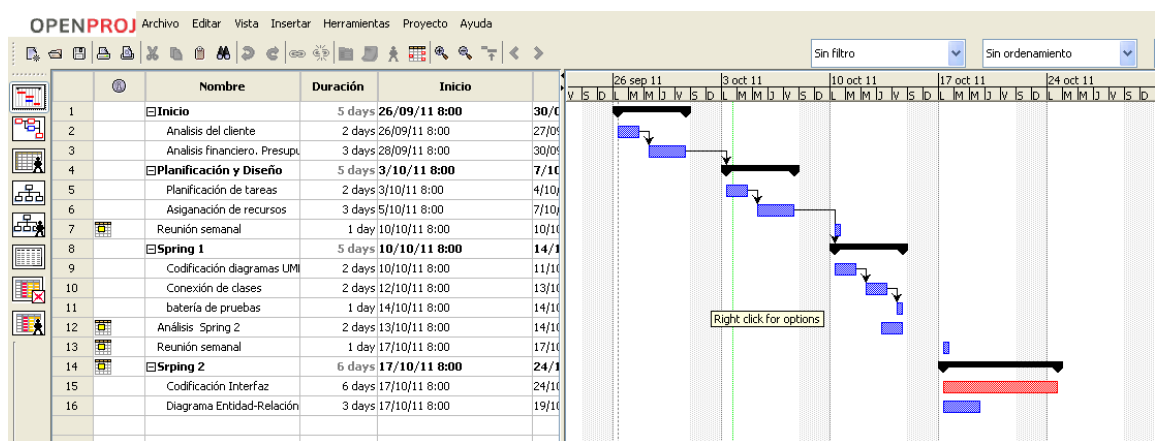


Ilustración 21. Diagrama de Gantt con OpenProject

OpenProject permite la asignación de recursos, tanto humanos como materiales, a las tareas, con el coste asociado que conlleva la explotación de dichos recursos.

También se permite la definición de calendarios laborales, incluyendo las horas que componen una jornada laboral, los días laborales o festivos.

⁶ El diagrama o gráfico Gantt es un herramienta útil en la gestión de proyectos, creada por Henry L. Gantt en 1917.

Consiste en confeccionar un cuadro con todas las actividades o tareas, por orden de inicio, con los respectivos tiempos previstos para su realización e identificación de la actividad precedente, a partir del cual se calculan las fechas de inicio y finalización, y se realiza una representación gráfica horizontal del comienzo y duración de todas las tareas del proyecto.

⁷ La Malla Pert es una herramienta cuantitativa de planificación y control, lo que permite a los administradores de proyectos contar con un modelo de optimización que entregue la solución óptima de una secuencia de actividades en el tiempo, que deben realizarse para finalizar el plan de acción. También permite al administrador programar un proyecto por adelantado y a la vez calcular el tiempo necesario para completarlo. Como herramienta de control, la Malla Pert facilita las actividades de control, permitiendo la comparación del tiempo real con el planificado.

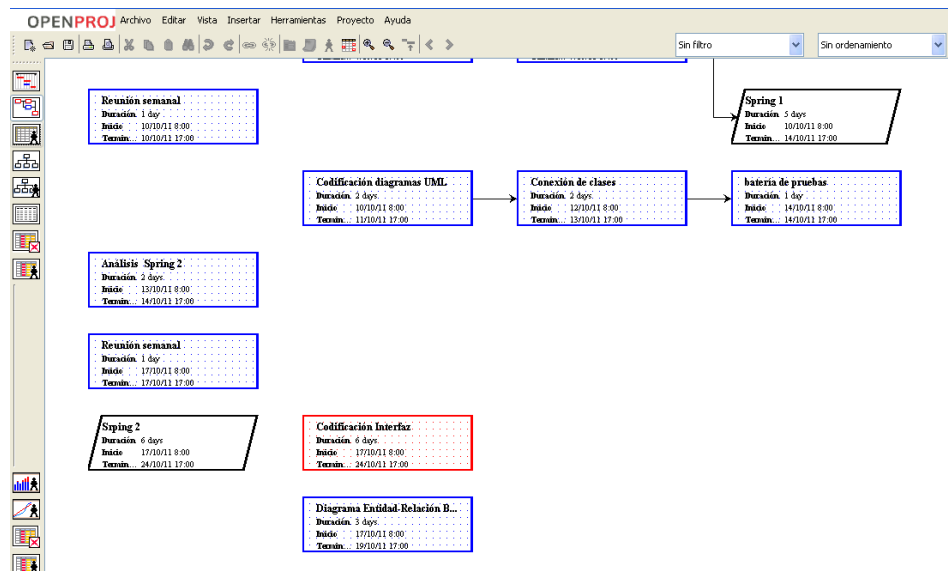


Ilustración 22. Diagrama de PERT con OpenProject

OpenProject permite al gestor de proyectos poder ver los costos que conlleva la utilización de los recursos en las tareas en las vistas “*detalle uso de tarea*” y “*detalle uso de recursos*”.

	Nombre	Tipo	Dirección de Correo elec...	Calendario Base	Tasa Estándar	Tasa sobretiempo	Costo Por Uso
1	Julio Ripoll	Trabajo	jripoll@alumnos.upm.es	default base	20/hora	35/hora	0,00 €
2	Maria Sáez	Trabajo	maria.saez@alumnos.upm.es	default base	18/hora	25/hora	0,00 €
3	Lucas Sanz	Trabajo	lucas.sanz@alumnos.upm.es	default base	18/hora	25/hora	0,00 €
4	Juan Perez	Trabajo	juan.perez@alumnos.upm.es	default base	12/hora	20/hora	0,00 €
5	Pedro León	Trabajo	pedro.leon@alumnos.upm.es	default base	12/hora	20/hora	0,00 €
6	Alicia Moreno	Trabajo	alicia.moreno@alumnos.up...	Cambio nocturno	16/hora	23/hora	0,00 €
7	Portatil HP	Material			0		15,00 €
8	Oficina	Material			17/hora		154,58 €

Ilustración 23. Gestión de recursos en OpenProject

Cada proyecto es guardado en un formato de archivo propio de la aplicación (.pod), permitiendo así crear un archivo inicial con la planificación del proyecto, y un segundo archivo con el desarrollo del proyecto, pudiendo realizar comparativas planificación/desarrollo real. Dichos archivos pueden ser compartidos mediante una unidad de red, siendo accesibles y modificables desde cualquier equipo que tenga instalado OpenProject. Si se produjese varios accesos al archivo contenedor del proyecto, únicamente podría ser modificado por el usuario que primero accediese a él, pudiendo el resto únicamente ver el contenido sin modificarlo.

También podemos realizar informes en PDF, HTML, CSV, XML y XLS y exportar e importar archivos con extensión .mpp (Microsoft Project).

Esta aplicación ha sido evaluada sobre una máquina virtual con sistema operativo Microsoft Windows XP. Para su instalación basta con descargar el instalador de la url <http://sourceforge.net/projects/openproj/> y seguir los pasos indicados en ella.

Esta aplicación ha sido desestimada para el propósito de este proyecto debido a que no permite una gestión multiproyecto simultánea con compartición de recursos. Pese a dicho inconveniente, esta aplicación es la recomendada en caso de no contar con varios proyectos en cartera, debido a su fácil manejo e instalación, su costo (es gratuita) y la posibilidad de gestionar desde la propia aplicación el costo del proyecto y sus tareas asociadas.

7.2. GanttProject:

Es una aplicación de escritorio para la programación y gestión de proyectos. Es multiplataforma y se puede instalar y ejecutar desde las plataformas Windows, Linux y MacOSX. Es gratuita y su código es opensource bajo licencia GPL 2.0 (General Public License, Version 2.0).

Con GanttProject podemos definir tareas, duraciones, hitos, dependencias, fechas de inicio y de fin de las tareas y de los proyectos, y días festivos. También podemos crear diagramas de Gantt, visionando la estructura del desglose de trabajo, dibujar dependencias y el cálculo de la ruta crítica. Permite realizar informes por tarea, por recurso, por día o semana.

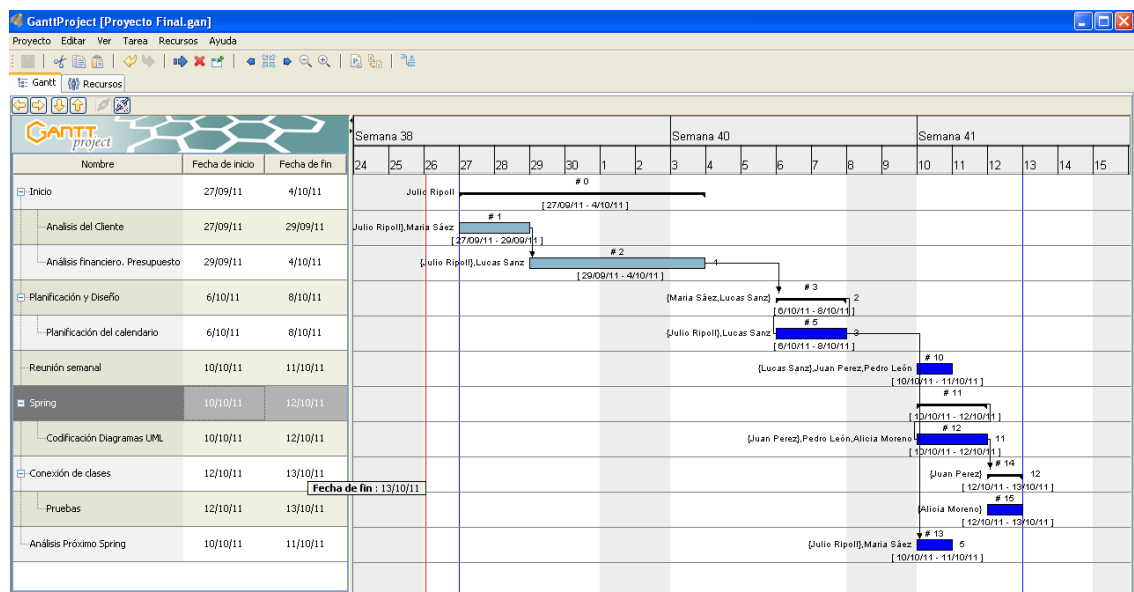


Ilustración 24. Diagrama de Gantt generado con Gannt Projct

Podemos asignar recursos tanto humanos como materiales a las tareas a realizar y visionarlos en la tabla de asignación de recursos, mostrando los recursos que están siendo sobreasignados. Como se puede observar en la Ilustración 25, se muestra cuando los recursos están siendo sobreexplotados y tienen asignadas más tareas de las que puede realizar en una jornada laboral definida con anterioridad.

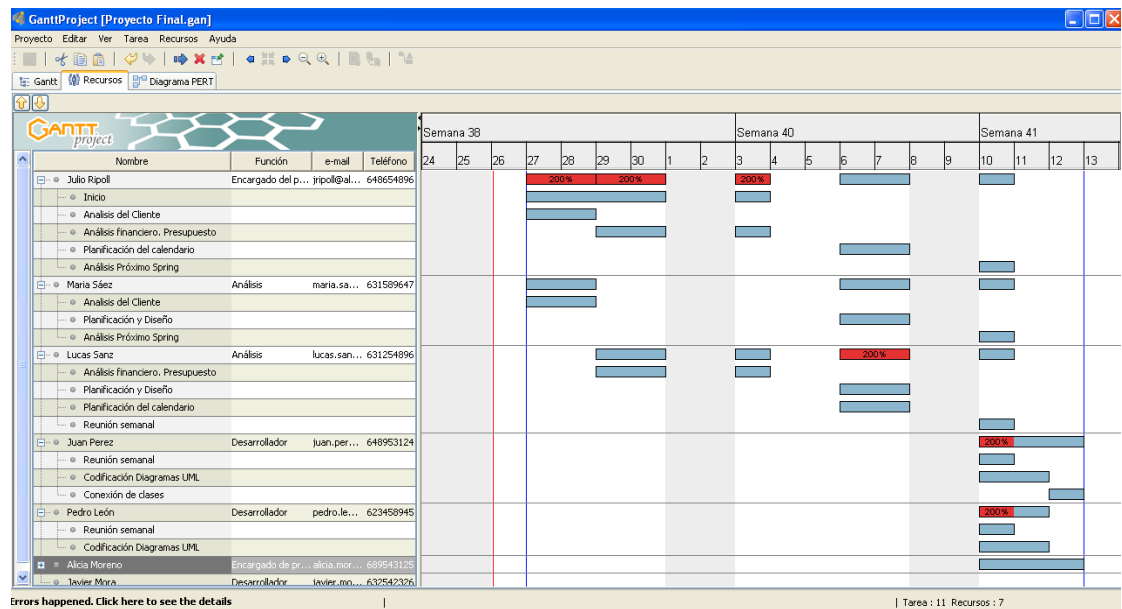


Ilustración 25. Asignación de recursos en GanttProject.

También se pueden generar diagramas PERT de los diagramas Gantt obtenidos anteriormente.

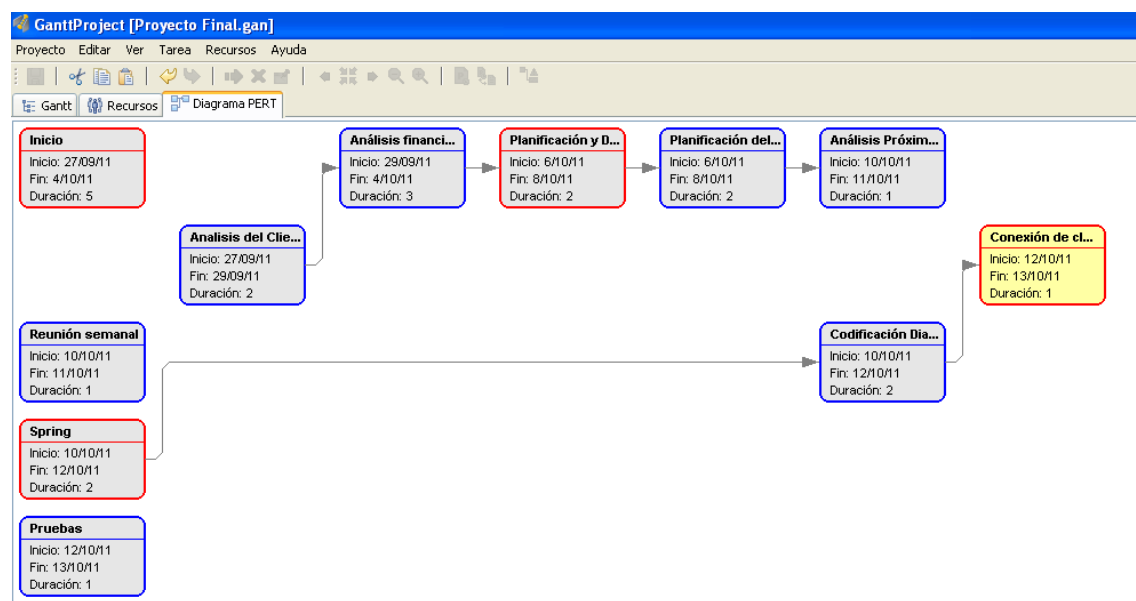


Ilustración 26. Diagrama de PERT generado con GanttProject.

Ganttproject utiliza para almacenar los proyectos un formato de archivo XML, lo que permite editarlo con cualquier editor de texto. Los archivos XML guardan

las listas de tareas, las listas de recursos y las relaciones entre las tareas y recursos. Este archivo usa las extensiones .xml o .gan. Permite exportar las gráficas a formato PNG, y los proyectos a los formatos .pdf y mediante tecnología XSL⁸ a formato .html, permitiendo al usuario crear su propia hoja de estilo XSL y así poder realizar la exportación de una forma personalizada. También podemos transmitir un proyecto a un servidor web y trabajar sobre el proyecto pudiendo compartir el proyecto utilizando WebDAV⁹.

Para poder hacer una diferenciación entre en la planificación inicial del proyecto y el desarrollo del mismo, bastaría con crear dos archivos y proceder a su comparación.

Esta aplicación ha sido evaluada sobre una máquina virtual con sistema operativo Microsoft Windows XP. Para su instalación basta con descargar el instalador de la url <http://www.ganttproject.biz/download> y seguir los pasos indicados en el asistente de instalación.

Esta aplicación ha sido desestimada para el propósito de este proyecto debido a que no permite una gestión multiproyecto simultánea con compartición de recursos.

⁸ XSL (Extensible Stylesheet Language, "lenguaje extensible de hojas de estilo") es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML debe ser transformada o formateada para su presentación.

⁹ WebDAV (Web-based Distributed Authoring and Versioning) es un conjunto de métodos basados en el Protocolo de transferencia de hipertexto (HTTP) que facilita la colaboración entre los usuarios de la edición y gestión de documentos y ficheros almacenados en servidores de la World Wide Web. WebDAV se definió en la RFC 4918 por un grupo de trabajo del Internet Engineering Task Force (IETF).

7.3. Project Planning and Tracking System (PPTS):

Aplicación web desarrollada para la gestión de proyectos, especialmente enfocada a las metodologías de gestión de proyectos ágiles Scrum y eXtreme Programming. PPTS es una aplicación de uso libre con licencia GNU (GPL).

Al tratarse de una aplicación web hay que destacar que trabaja sobre una base de datos MySQL, permitiendo un manejo de datos de forma muy personalizada, y un servidor web, en nuestro caso Apache. Cuenta con una avanzada gestión de usuarios y de roles de acceso a la aplicación, los cuales pueden ver sus asignaciones a las tareas del proyecto, días festivos, reserva de salas para las reuniones.

Permite una clara administración de las iteraciones, pudiendo ver de forma clara que tareas son propias de cada sprint.

Project Planning & Tracking System v. 1.6.6

Home My... Iteration Backlog ProjectManager Management Admin

Personal Report

Select a report:

Week	Change week	User	Options
2011-10-24 View	<< < > >>	First user of PPTS	<input type="checkbox"/> Show current project only <input type="checkbox"/> Calculate ToGo hours automatically

Overview:

[Generate PDF](#)

Proyecto Final PFC										
Análisis viabilidad del proyecto	Mon 2011-10-24		Tue 2011-10-25		Wed 2011-10-26		Thu 2011-10-27		Fri 2011-10-28	
	Used	ToGo	Used	ToGo	Used	ToGo	Used	ToGo	Used	ToGo
1. Estudio Financiero										
• Estudio balance anual										
• Estudio ingresos brutos										
2. Analisis Propuesta de Software										
• Reunión con cliente										
Daily Totals:										
Weekly Total Used:										0

Ilustración 27. Planificación de tareas en PPTS

PPTS también permite ver una clara asignación de recursos por fechas, mostrando que carga posee el recurso, incluso marcando como sobreasignados en caso de que se produjese.

7.4. ScrumWorks:

Aplicación desarrollada en Java por la empresa CoballNet para apoyar a las empresas a la transición a las metodologías ágiles de gestión de proyectos. Está especialmente diseñada para trabajar con la metodología ágil Scrum, aumentando la transparencia y claridad del proyecto y promoviendo la colaboración del equipo de desarrollo, disminuyendo el riesgo del proyecto y los costes de desarrollo. Cuenta con una gestión avanzada de usuarios y roles y debido a su desarrollo en Java es accesible desde cualquier sistema operativo. Cuenta con dos entornos de trabajo, un acceso vía web y vía escritorio.

7.4.1. Entorno Web:

Para poder acceder al entorno web debemos tener la aplicación instalada en un servidor, pudiendo estar en la intranet de la empresa, como en un servidor externo o en la nube, y acceder a la url sobre la que se ha instalado la aplicación. Una vez se ha accedido al entorno web podemos encontrar varias secciones bajo las que podemos trabajar.

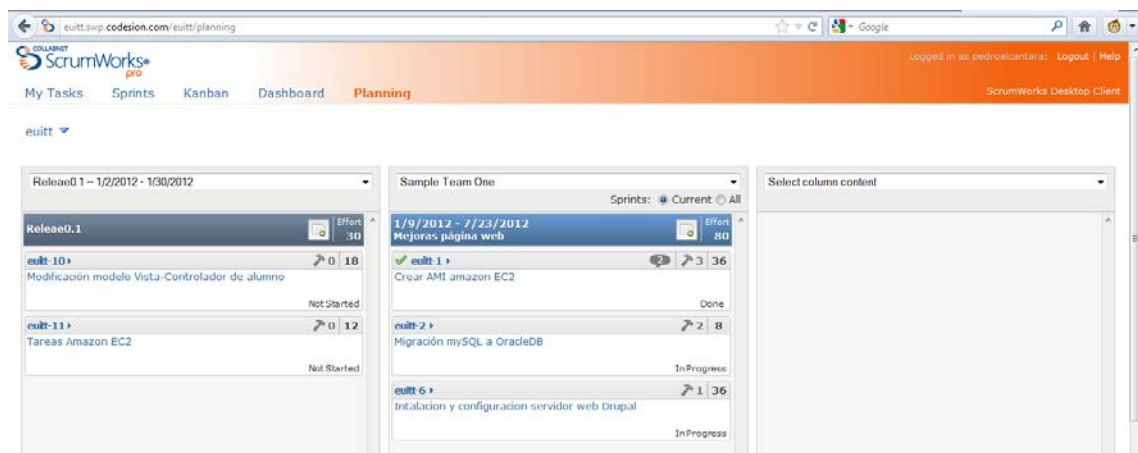


Ilustración 29 Planificación de proyectos en ScrumWorks

En esta vista podremos gestionar la planificación de los diferentes proyectos que tengamos en cartera (Planning), acceder a los reportes generados con la información del proyecto seleccionado (Dashboard), visualizar las tareas pendientes (My Tasks), la información de los Sprints de los proyectos (Sprints),

y visualizar en que parte del flujo de trabajo se encuentran las tareas del proyecto (Kanban¹⁰).

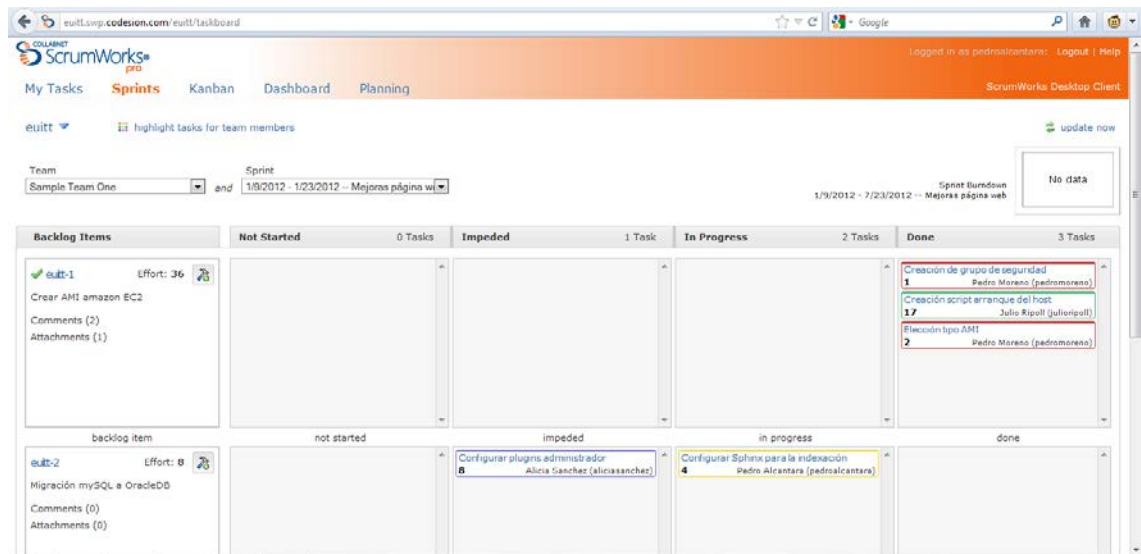


Ilustración 30 Tareas a realizar en el Spring

7.4.2. Entorno Escritorio

Desde el entorno de escritorio se pueden acceder a 3 apartados.

El primero es “Product Backlog”, o pila de tareas del producto, el cual representa un conjunto de tareas que puede ser completado por el equipo en la duración de un Sprint.

euit: Product Backlog						
Key	Uncommitted Backlog Items/Tasks	Backlog Effort	Task Hours	Status	rBV	ROI
	Release0.1	Total: 86	Total: 32			
euit-1	[AMIs Amazon EC2] Crear AMI amazon EC2	36		Not Started	12%	0,32
	Elección tipo AMI		2	Done		
	Creación de grupo de seguridad		1	Done		
	Creación script arranque del host		17	Done		
euit-2	Migración MySQL a OracleDB	8		Not Started	38%	4,65
	Configurar Sphinx para la indexación		4	In Progress		
	Configurar plugins administrador		8	Impeded		
euit-6	[Drupal en preproduc...] Instalación y configuración servidor web Drupal	24		Not Started	23%	0,92
	Instalación Drupal 7		0	Done		
euit-10	Modificación modelo Vista-Controlador de la clase Alumno	18		Not Started	28%	1,51

Ilustración 31 Pila de tareas

¹⁰ Kanban: Palabra japonesa que puede traducirse como “tarjetas visuales” (“Kan” visual y “Ban” tarjeta) y su objetivo en el desarrollo de software es, visualizar en una pizarra o tablón, como se van completando las tareas de un spring, mientras desarrollamos sobre metodologías ágiles, a través del flujo de trabajo, compuesto por tareas a realizar o backlog, tareas seleccionadas, desarrollo de tareas (en proceso o realizada), pruebas y terminada.

El segundo elemento es “Release Planer”, el cual nos proporciona una vista con la evolución que están teniendo los equipos de desarrollo con las tareas que componen el Sprint, con el estado en el que se encuentran y la dedicación en horas realizada por cada desarrollador.

Key	Committed Backlog Items/Tasks	Backlog Effort	Task Hours	Status
	Sprint -- 9.1.2012 - 23.1.2012 -- Mejoras página web	Total: 98	Total: 28	
euit-1	✓ [AMIs Amazon EC2] Crear AMI amazon EC2	36		Done
	Elección tipo AMI		2	Done
	Creación de grupo de seguridad		1	Done
	Creación script arranque del host		17	Done
euit-2	✗ Migración MySQL a OracleDB	8		In Progress
	Configurar Sphinx para la indexación		0	Done
	Configurar plugins administrador		8	Impeded
euit-6	✗ [Drupal en preproduc...] Instalación y configuración servidor web Drupal	24		In Progress
	Instalación Drupal 7		0	Done
euit-10	✓ ✗ Modificación modelo Vista-Controlador de la clase Alumno	18		Done
euit-11	✗ [AMIs Amazon EC2] Tareas Amazon EC2	12		In Progress

Ilustración 32 Planificación del Spring

Key	Backlog Items by Release/Epic	Backlog Effort	Status
	Release0.1 2.1.2012 - 30.1.2012	Total: 98	
	✗ AMIs Amazon EC2 Budget: 48	Total: 48	
euit-1	Crear AMI amazon EC2	36	Done
euit-11	Tareas Amazon EC2	12	In Progress
	✗ Drupal en preproducción Budget: 28	Total: 24	
euit-6	Instalación y configuración servidor web Drupal	24	In Progress
	✗ Uncategorized	Total: 26	
euit-2	Migración MySQL a OracleDB	8	In Progress
euit-10	Modificación modelo Vista-Controlador de la clase Alumno	18	Done

Ilustración 33 Evolución del Spring

Por último podemos acceder al apartado de reportes, tanto del proyecto como del Sprint, los cuales nos permiten ver cómo evolucionan las tareas, las estimaciones de tiempo contra el desarrollo real de las tareas, la velocidad del equipo y horas invertidas en cada ticket.

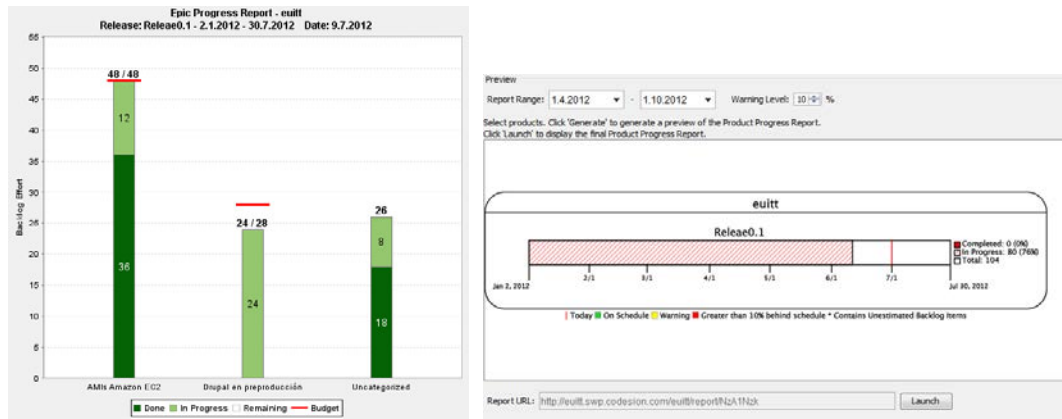


Ilustración 34 Reportes de evolución del Spring

Para la evaluación de la herramienta se ha realizado en una máquina virtual con sistema operativo OpenSuse, pero debido a su acceso web y su entorno de escritorio desarrollada en tecnología Java, se puede acceder desde cualquier sistema operativo con entorno visual.

Al ser una aplicación de pago se ha trabajado con una versión de evaluación totalmente funcional pero limitada en tiempo de uso a 30 días naturales. Para poder disfrutar de dicha aplicación basta con visitar la página web de Collab, y en la sección de descargas¹¹ proceder a registrarse. Nos facilitarán una url de acceso para la evaluación del entorno web de la aplicación. Para acceder al entorno de escritorio basta con descargar desde el enlace denominado “ScrumWorks Desktop Client” que se encuentra en el panel principal del entorno web. Una vez descargado basta con ejecutar el archivo, introducir el nombre de usuario y la contraseña y ya tendremos acceso al entorno de escritorio de nuestros proyectos ya que la aplicación lleva la configuración necesaria para conectarnos a nuestro servidor.

Esta aplicación cumple con los requisitos que se han estimado para este proyecto, pero ha sido desestimada porque únicamente permite la gestión de proyectos aplicando la metodología ágil Scrum, dejando al resto de metodologías de lado.

¹¹ <http://www.collab.net/downloads/scrumworks>

7.5. JIRA Issue and project tracking:

Aplicación web desarrollada, mediante tecnología Java, por la compañía australiana Atlassian. JIRA pretende ser una aplicación con la que poder gestionar proyectos, el seguimiento de errores y el seguimiento de problemas, y así poder mejorar la calidad del código y la velocidad de desarrollo de los equipos de desarrollo.

La principal característica de JIRA es lo extensible que es, ya que cuenta con una gran cantidad de extensiones¹² con las que nos permite añadir funcionales tales como análisis de calidad de código, control de versiones, gráficas de los flujos de trabajo, herramientas de comunicación para los equipos de desarrollo, etc., lo que lo convierte en una herramienta de gestión de proyectos muy potente.

JIRA cuenta con una gestión de permisos de usuarios, grupos y funciones, para restringir el acceso de contenidos y acciones a cualquier nivel de los proyectos.

Permite la centralización de todos los proyectos y componentes, creando una visión general de la cartera de proyectos actual, pudiendo realizar un análisis de requisitos, definir las tareas y programarlas en función del plazo de entrega o de un hito, o modificar su prioridad durante el desarrollo del proyecto.

A las tareas se les puede adjuntar archivos, con especificaciones, requisitos, ayudas o lo que se desee.



Mostrando incidencias 1 a 3 de 3 incidencias encontradas

T	Clave	Sumario	Responsable	Informante	Pr	Estado	Res	Creado	Actualizado	Fecha de Entrega
	JRIPOLLUPM-3	Codificación GUI	Pedro Moreno	Julio Ripoll	↑	Open	Sin resolver	09/oct/11	09/oct/11	18/oct/11
	JRIPOLLUPM-2	JRIPOLLUPM-1 / Estudio Api Google	Alicia Moreno	Julio Ripoll	↓	Resolved	Fixed	06/oct/11	09/oct/11	07/oct/11
	JRIPOLLUPM-1	Integración con API de Google	Pedro Moreno	Julio Ripoll	↑	Resolved	Fixed	06/oct/11	09/oct/11	16/nov/11

Mostrando incidencias 1 a 3 de 3 incidencias encontradas

Ilustración 35. Panel individual para la gestión de tareas.

¹² Portal de acceso a las extensiones para JIRA: <https://marketplace.atlassian.com/>

JIRA cuenta con gran cantidad de reportes, tales como el tiempo transcurrido del sprint, la carga de trabajo por usuario o métricas de las tareas, y así poder seguir con claridad el desarrollo del proyecto y sus incidencias. Toda esta información es accesible desde la pizarra que dispone cada usuario de JIRA. Dicha pizarra, es totalmente personalizable mediante gadget.



Ilustración 36 Pizarra de reportes.

A parte de la funcionalidad de reportes de estados de los gadgets, cabe destacar el apoyo que realiza a la comunicación, permitiendo a los miembros de un equipo o entre diferentes equipos una comunicación muy fluida y esencial en el desarrollo ágil de software a través de paneles de comunicación, foros y notificaciones vía email que nos ofrece JIRA.

La aplicación también nos ofrece una clara visualización del estado de los recursos, su asignación a tareas, ver el desarrollo del flujo de trabajo y los inconvenientes surgidos en el Sprint y el proyecto, permitiendo modificar la prioridad, los recursos asignados y el esfuerzo dedicado a las tareas.

Por último podemos destacar que los proyectos pueden ser exportados a archivos tipo Word, Excel, RSS o XML y que permite trabajar sobre una base de datos en local o en un servidor remoto.

Esta aplicación ha sido evaluada en una máquina virtual con sistema operativo Microsoft Windows XP y con una versión de prueba¹³ con todas las funcionalidades, pero limitado a 30 días de uso. JIRA funciona bajo un servidor de aplicaciones web, que en nuestro caso viene incluido en el instalador de JIRA. En caso de quererse instalar JIRA en otro sistema operativo, bastará con tener instalado un servidor web de aplicaciones (tomcat o jetty), descargarse el archivo con extensión war y alojarlo en el directorio correspondiente.

JIRA finalmente ha sido descartada por encontrar otras aplicaciones de gestión de proyectos que se adaptan mejor desde un inicio a la gestión ágil de proyectos. De todos modos JIRA se recomienda en caso de necesitar una aplicación de gestión de proyecto con características muy concretas que los complementos anteriormente mencionados sean capaces de satisfacer.

¹³ <http://www.atlassian.com/software/jira/download>

7.6. DotProject:

Es una aplicación web de libre distribución y código abierto para la gestión y organización de proyectos. Tiene como fin proporcionar al gestor de proyectos una herramienta con la que poder administrar las tareas, los recursos, los calendarios y la comunicación de los equipos de desarrollo que se disponga.

Actualmente se encuentra en su versión 2.1.5 y al ser una aplicación web cuenta con características propias de dichas aplicaciones, como son una avanzada gestión de usuarios y sus roles, administración de módulos sobre el software base, importación y exportación de ficheros en el servidor, personalización de menús e idiomas, instalación en diferentes sistemas operativos, etc.

DotProject consta de módulos ordenados jerárquicamente, los cuales permiten dar funcionalidad a la aplicación. Los principales módulos de los que consta Dotproject son:

- Las compañías, las cuales constan de proyectos, actividades y usuarios.

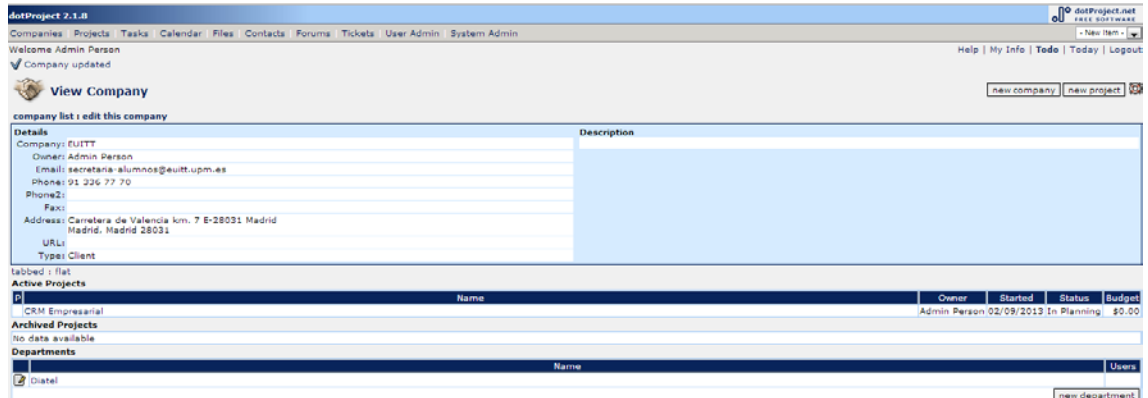


Ilustración 37 Vista de una compañía en DotProject

- Los departamentos: son áreas que componen las compañías, los cuales agrupan usuarios en este nivel.
- Los usuarios, que son capaces de acceder a dotProject y realizar las tareas que el sistema de permisos concede a su rol.
- Los contactos, son usuarios especiales que pueden asignarse a un proyecto pudiendo recibir correos, actualizaciones y noticias del proyecto asignado. No es imprescindible que tenga un usuario asociado.

- Los costos, permite establecer los costos del proyecto de mano de obra, de materiales, de administración de riesgo, de infraestructura (edificios, máquinas, etc.), equipos y utilidades.
- Los proyectos, que contienen el grupo de tareas necesarias para desarrollar un determinado producto o servicio.

View Project

projects list : edit this project : organize tasks : reports

CRM Empresarial

Details

Company: EUITT
 Internal Company:
 Short Name: CRM Empres
 Start Date: 02/09/2013
 Target End Date: 30/09/2013
 Actual End Date: 30/09/2013
 Target Budget: \$0.00
 Project Owner: Person: Admin
 URL:
 Staging URL:

Summary

Status: In planning
 Priority: normal
 Type: Unknown
 Progress: 6.8%
 Worked Hours:
 Scheduled Hours: 41
 Project Hours: 41

Description

tabbed : flat

Tasks Tasks (Inactive) Forums Gantt Chart Task Logs Events Files

Expand All/One Level Collapse All/One Level

Icon	Task Name	Task Owner	Assigned Users	Start Date	Duration	Finish Date
Log 10%	Task1 - Creación proyecto	admin	admin (100%)	03/09/2013 08:15 am	1 hours	20/09/2013 08:15 am
Log 20%	Task2 - Creación entornos de desarrollo	admin	admin (100%)	08/09/2013 08:15 am	8 hours	04/09/2013 08:15 am
Log 0%	Task3 - Creación modelo de dominio Usuarios	admin	admin (100%)	09/09/2013 08:30 am	16 hours	11/09/2013 08:45 am
Log 0%	Definición y creación de roles	admin	admin (100%)	09/09/2013 08:30 am	8 hours	10/09/2013 08:45 am
Log 0%	Mapeo permisos de rol-role	admin	admin (100%)	10/09/2013 08:30 am	8 hours	11/09/2013 08:45 am

Summaries

Key: #Future Task #Started and on time #Should have started #Overdue #Done Expand All/One Level Collapse All/One Level

Ilustración 38 Vista de proyecto de DotProject

- Las actividades, que son las tareas asignadas a un proyecto, de las cuales se puede establecer la duración, las dependencias, los recursos asignados y su progreso
- Diagramas de Gantt, los que nos permiten ver de forma gráfica las tareas del proyecto.
- Los tickets, que permiten administrar los problemas asociados a un proyecto.

DotProject admite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos. También permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto y también permite asignar recursos no humanos (oficinas, equipamiento, etc.) a un proyecto.

Esta aplicación ha sido evaluada sobre una máquina virtual que implementa como sistema operativo OpenSuse. Se ha instalado en un servidor web formado por Apache, MySQL y PHP.

DotProject ha sido desestimada debido a la dificultad que se tiene de adaptar la aplicación a las metodologías ágiles y sobre todo por las imposiciones de campos obligatorios de fechas de inicio y fin de tareas, impidiendo dar flexibilidad a las horas dedicadas a las tareas planificadas.

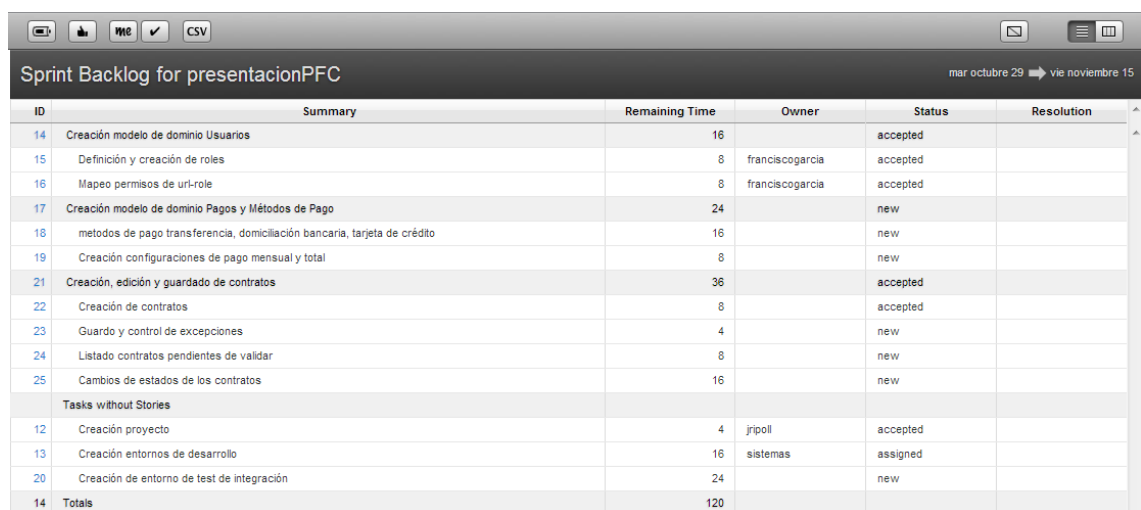
7.7. Agilo Trac

Aplicación web de libre distribución basada en la aplicación Trac, especialmente desarrollada para poder soportar metodologías ágiles de gestión de proyectos. Su primera versión fue lanzada en 2008 y todas ellas se encuentran desarrolladas en python y funcionan bajo un servidor web Apache.

Al ser una aplicación web permite una gestión avanzada de usuarios, permitiendo definir la duración de la jornada laboral, los días de trabajo y vacaciones, y asignarle a un equipo de trabajo. En los equipos de trabajo permite añadir tantos usuarios como se desee, haciendo un resumen de la capacidad del equipo semanal en horas.

Agilo Trac está enfocado a una gestión simple del desarrollo de software, por ello permite organizar los proyectos atendiendo a las metodologías ágiles de gestión de proyectos Extrem Programming y Scrum.

Con esta aplicación podemos ir creando, en primer lugar historias de usuarios, para posteriormente ir añadiéndole tareas, llenando el Product Backlog, con sus estimaciones de tiempo, propietario de la tarea o historia, estado... y si se desea, más características ya definidas por la aplicación y/o campos personalizados (tipo texto, numérico, listas y seleccionables), con las cuales podremos ir creando la próxima iteración del desarrollo del software.



ID	Summary	Remaining Time	Owner	Status	Resolution
14	Creación modelo de dominio Usuarios	16		accepted	
15	Definición y creación de roles	8	franciscogarcia	accepted	
16	Mapeo permisos de url-role	8	franciscogarcia	accepted	
17	Creación modelo de dominio Pagos y Métodos de Pago	24		new	
18	metodos de pago transferencia, domiciliación bancaria, tarjeta de crédito	16		new	
19	Creación configuraciones de pago mensual y total	8		new	
21	Creación, edición y guardado de contratos	36		accepted	
22	Creación de contratos	8		accepted	
23	Guardo y control de excepciones	4		new	
24	Listado contratos pendientes de validar	8		new	
25	Cambios de estados de los contratos	16		new	
Tasks without Stories					
12	Creación proyecto	4	jripoll	accepted	
13	Creación entornos de desarrollo	16	sistemas	assigned	
20	Creación de entorno de test de integración	24		new	
14	Totals	120			

Ilustración 39. Historias de usuario y tareas

Agilo trac permite una vez comenzado la iteración, trabajar o bien con el Sprint Backlog o con el tablero de Kanban, permitiendo en las reuniones diarias de seguimiento imputar las horas dedicadas a cada tarea y modificar su estado.

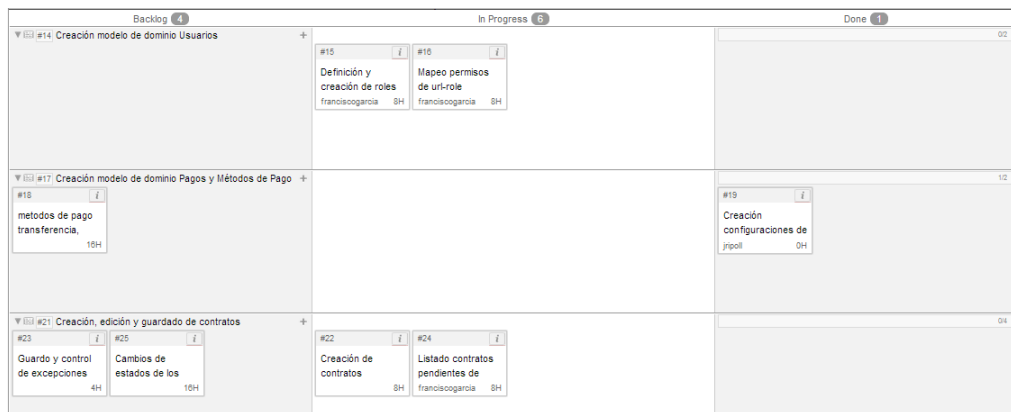


Ilustración 40. Tablero Kanban

También permite ver reportes típicos de XP y Scrum, como puede ser el Burndown Chart, el cual representa el trabajo pendiente a realizar a lo largo de una línea temporal. En la gráfica se representa, la fecha de inicio, la fecha estimada de finalización, la fecha real de estimación, la evolución de la carga de trabajo y el tiempo total de la iteración del equipo de desarrollo.



Ilustración 41. Trabajo pendiente en el Sprint

También podemos obtener gráficos y reportes de la situación actual de las tareas del sprint y así saber cómo va evolucionando los desarrollos planificados.

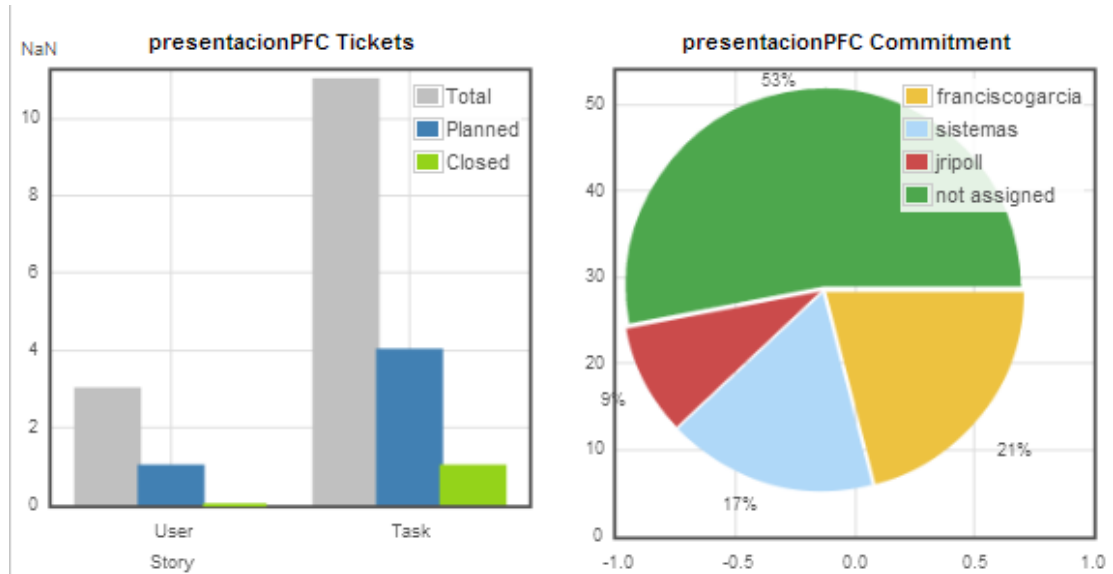


Ilustración 42. Tareas planeadas/cerradas y asignación de tareas

Agilo Trac también nos permite definir la dedicación de cada recurso a los proyectos, medido en horas, y si se está realizando una sobreasignación de tareas al recurso, es decir, asignando más horas de trabajo por día que la capacidad de la que dispone el recurso.

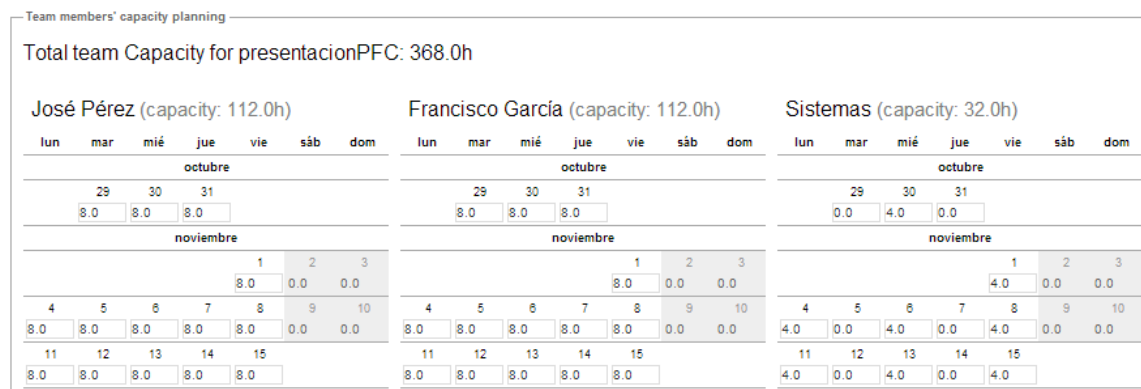


Ilustración 43. Definición de capacidad de cada recurso

Name	04/11/2013	05/11/2013	06/11/2013	07/11/2013	08/11/2013	09/11/2013	10/11/2013	11/11/2013
Francisco García	20.0h	20.0h	20.0h	20.0h	20.0h	20.0h	20.0h	20.0h
Sistemas	16.0h	16.0h	16.0h	16.0h	16.0h	16.0h	16.0h	16.0h
Julio Ripoll	8.0h	8.0h	8.0h	8.0h	8.0h	8.0h	8.0h	8.0h
not assigned	50.0h	50.0h	50.0h	50.0h	50.0h	50.0h	50.0h	50.0h
Total	94.0	94.0	94.0	94.0	94.0	94.0	94.0	94.0

Ilustración 44. Sobreasignación de tareas a recursos

De Agilo Trac también podemos destacar su capacidad de:

Establecer notificaciones vía correo electrónico por cambios realizados sobre usuarios y tareas.

Dispone de wiki o base de conocimiento separada por proyectos.

Permite la exportación tanto a ficheros Excel como a ficheros CSV.

Puede ser integrado con el controlador de versionado del software Subversion.

7.7.1. Instalación Agilo Trac

Actualmente la aplicación está disponible para su instalación de cuatro maneras diferentes:

- Windows Installer: Utilizando el motor de instalación, mantenimiento y eliminación de aplicaciones de Microsoft Windows
- Máquina virtual: Máquina virtual que utiliza tecnología VM Ware, la cual permite arrancar dicha máquina y disponer de la aplicación sin necesidad de haber realizado ninguna instalación en equipo alguno.
- Python Eggs: Módulos de Python empaquetados con diversa información, como los módulos de los que depende la aplicación, la documentación relativa al software o la versión de la que se trata.
- Servidor en la nube: La empresa Agilo Software provee de un servidor en la nube. Su precio puede variar en función del número máximo de usuarios y la duración del servicio, pudiéndose encontrar en la siguiente url <http://agilo.enstore.com/item/hosted-service>.

Para la realización del análisis de la aplicación se ha decidido instalar la versión Windows Installer, la cual requiere de una mínima configuración para poder disfrutar de la aplicación. Durante la instalación basta con seleccionar los componentes que se desean instalar:

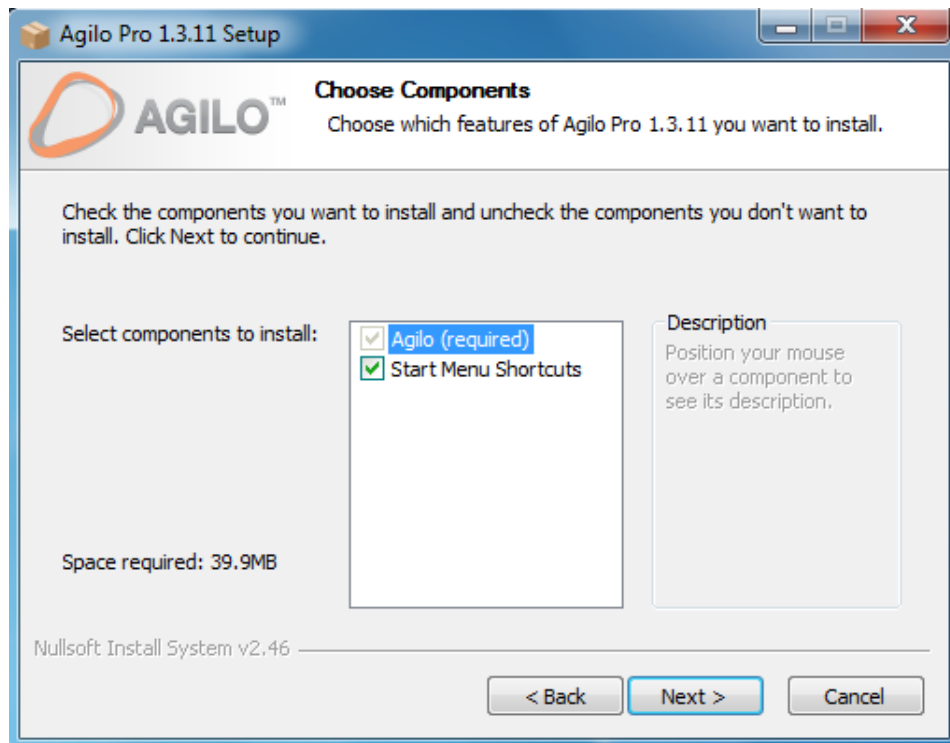


Ilustración 45. Módulos a instalar en Agilo

En nuestro caso elegiremos tanto Agilo, como los menús de inicio, los cuales nos permiten arrancar el servidor web con la aplicación y poder acceder a ella de una manera rápida y sencilla.

Tras esto nos preguntará en que directorio se desea instalar, se selecciona el deseado y se comienzan a instalar todos los paquetes necesarios hasta finalizar satisfactoriamente la instalación.

7.8. Rational Plan Multiproject:

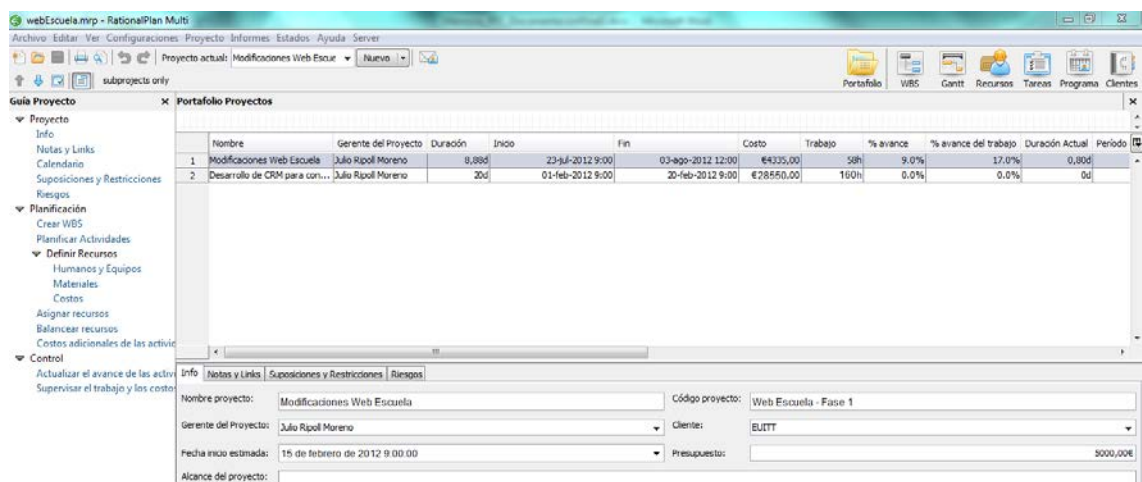
Aplicación de escritorio desarrollada con tecnología Java por la empresa Stand By Soft Ltd. para los administradores de proyectos con el objetivo de ayudarles en el desarrollo de la planificación, la asignación de los recursos disponibles, el seguimiento de las tareas y sus progresos, la administración de los presupuestos y en el análisis de la carga de trabajo de los recursos. Actualmente se encuentra en la versión 4.0, la cual es compatible con cualquier sistema operativo con tecnología Java, y en la se dispone de una versión de prueba totalmente funcional y limitada en el periodo de prueba.

Rational Plan Multiproject está especialmente diseñado y recomendado para la gestión de varios proyectos interrelacionados, ya sea por dependencia de tareas entre los diferentes proyectos o por contar con recursos compartidos.

Como principales características de Rational Plan Multiproject podemos destacar los siguientes puntos:

1. Portafolio del proyecto:

Permite definir los proyectos de forma independiente o con una estructura de árbol, pudiendo crear proyectos anidados. Desde la vista de Portafolio podemos observar los datos fundamentales de la cartera de proyectos



Nombre	Gerente del Proyecto	Duración	Inicio	Fin	Costo	Trabajo	% avance	% avance del trabajo	Duración Actual	Periodo
1 Modificaciones Web Escuela	Julio Ripoll Moreno	8,88d	23-jul-2012 9:00	03-ago-2012 12:00	€4335,00	38h	9.0%	17.0%	0,88d	
2 Desarrollo de CRM para con...	Julio Ripoll Moreno	20d	01-feb-2012 9:00	20-feb-2012 9:00	€78550,00	160h	0.0%	0.0%	0d	

Nombre proyecto:	Modificaciones Web Escuela	Código proyecto:	Web Escuela - Fase 1
Gerente del Proyecto:	Julio Ripoll Moreno	Cliente:	BUIT
Fecha inicio estimada:	15 de febrero de 2012 9:00:00	Presupuesto:	5000,00€
Alcance del proyecto:			

Ilustración 46 Portafolio de proyectos

2. Planificación del proyecto:

Rational Plan Multiproject permite crear una clara estructura de las tareas a realizar por cada proyecto, las dependencias existentes entre ellas, establecerles la fecha de inicio, de fin, retraso, el avance realizado y los costes asociados.

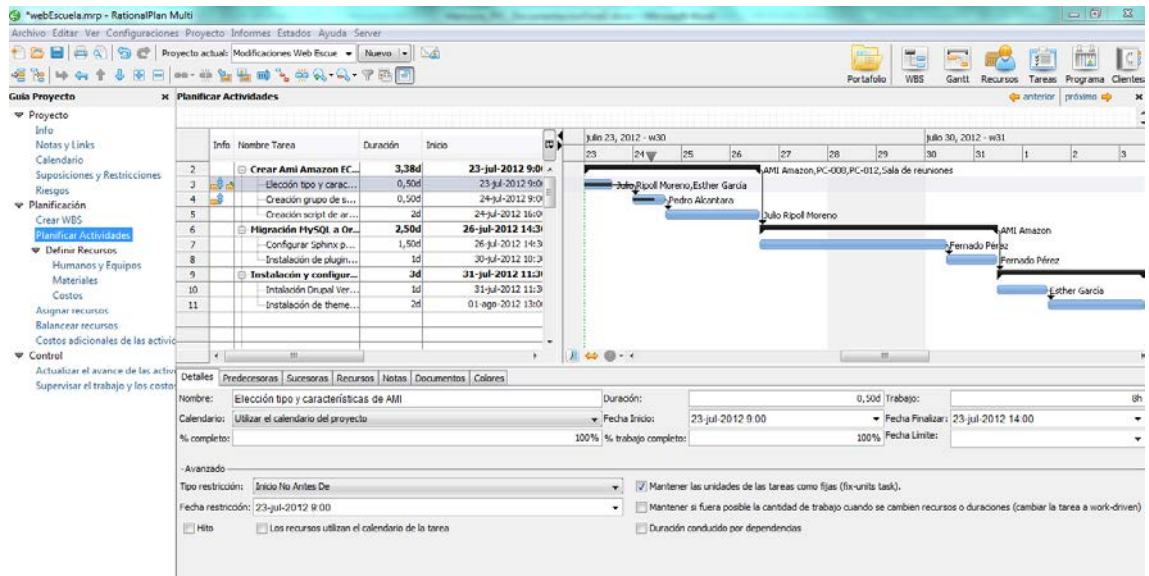


Ilustración 47 Planificación de tareas de un proyecto.

3. Administración de los recursos:

La aplicación permite definir recursos tanto humanos, como materiales y ser compartidos entre los diferentes proyectos que se tengan en cartera.

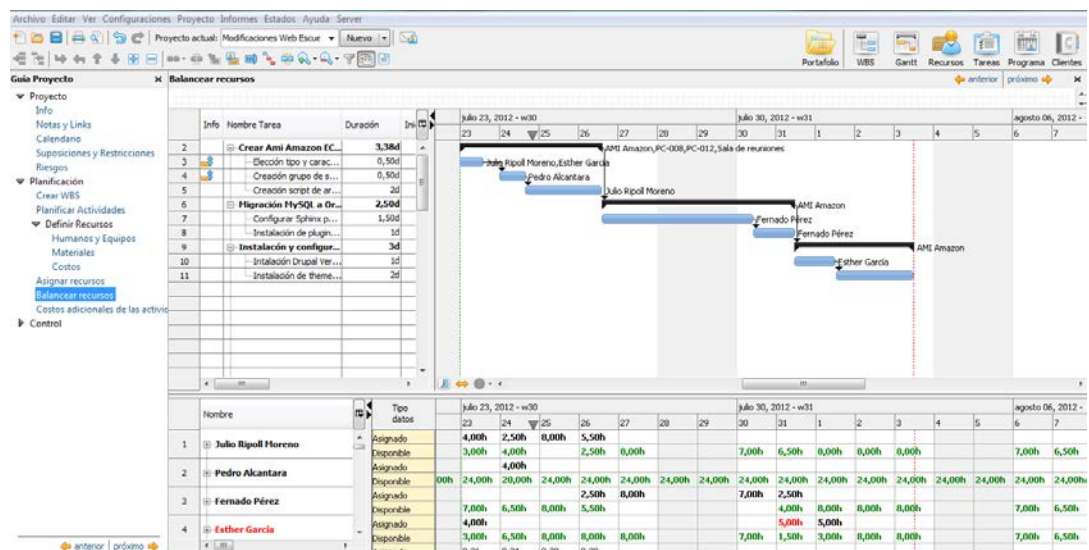


Ilustración 48 Gestión de recursos.

Respecto a los recursos humanos permite definir la duración de cada jornada laboral, ver si los recursos están sobreasignados o si han realizado horas extraordinarias. También permite ver el tiempo que el recurso ha invertido en cada tarea y el costo asociado.

4. Seguimiento de los proyectos:

Esta aplicación permite visualizar las tareas críticas del proyecto en función de una desviación ajustable y previamente definida. También se puede ir cambiando el progreso de las tareas, observando claramente cómo va avanzando la tarea hasta su consecución, pudiendo analizar el tiempo invertido y el costo derivado de su realización.

Como otras características destacables podemos encontrar:

- Gráficos de Gantt fáciles de usar.
- Existencia de una guía paso a paso con explicaciones que permite crear proyectos de forma fácil e intuitiva.
- Permite interoperabilidad con Microsoft Project, guardando los proyectos con formato xml, y con Microsoft Excel.
- Permite la realización de informes imprimibles y en formato PDF.
- Puede obtenerse la versión servidor, la cual permite tener los proyectos en una base de datos centralizada y que los usuarios accedan de forma remota desde sus equipos personales a través de la aplicación de escritorio o de un servidor Web.

7.8.1. Instalación de Rational Plan Multiproject:

Rational Plan Multiproject no requiere de instalación, simplemente hace falta tener instalado Java en nuestro equipo, ya que Stand By Soft Ltd desde su página web (<http://www.rationalplan.com/download.php>), y tras realizar el registro, nos facilitará un archivo con extensión .jar, el cual simplemente hay que ejecutar para poder tener acceso a la aplicación.

7.9. Microsoft Project Server 2007

Microsoft Project Server es la aplicación desarrollada por Microsoft Corporation con tecnología .NET para dar solución a la gestión de proyectos. La primera aplicación de gestión de proyectos desarrollada por Microsoft fue publicada en 1984 para el sistema operativo DOS. Se fueron desarrollando sucesivas versiones hasta la primera versión de Microsoft Project para el sistema operativo Windows en 1990. La primera versión desarrollada para servidores se lanzó en el año 2000 bajo el nombre de Microsoft Central, evolucionando hasta la versión aquí evaluada.

Microsoft Project Server es la aplicación más compleja y completa de las evaluadas, pudiéndosela considerar un conjunto de aplicaciones que cooperan para ofrecer al usuario final una completa herramienta de gestión de proyectos.

Esta herramienta puede dividirse en tres grandes niveles:

- Nivel web de Office Project Server 2007
- Nivel de aplicación de Office Project Server 2007
- Nivel de base de datos de Office Project Server 2007 (Microsoft SQL Server 2003)

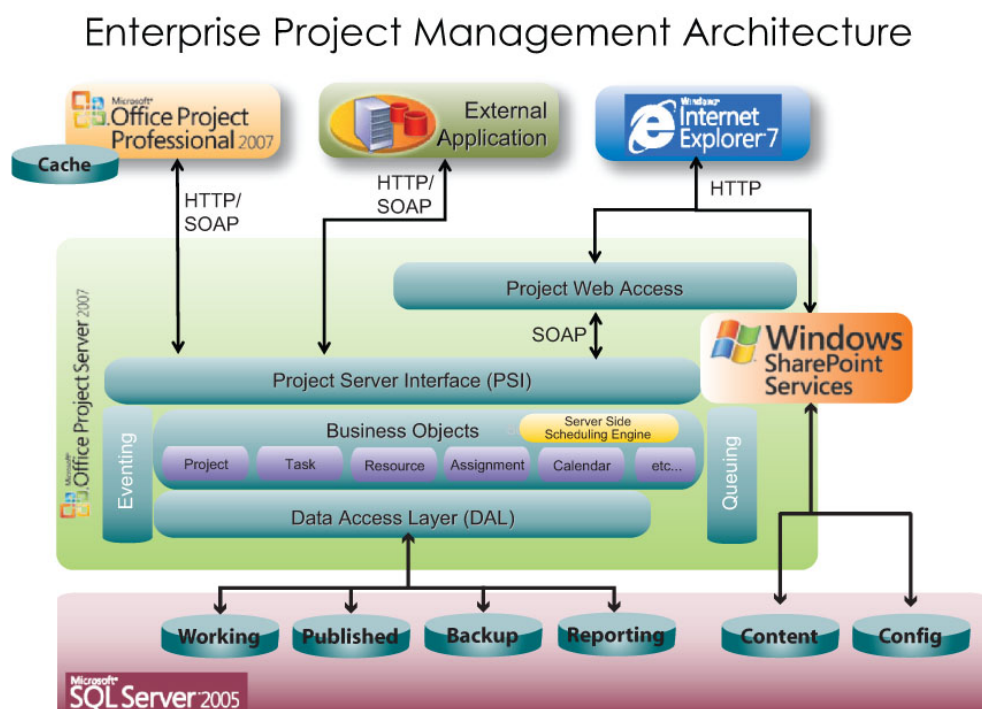


Ilustración 49 Arquitectura de Microsoft Project Server.

7.9.1. El nivel web de Office Project Server 2007 incluye dos componentes:

1. Microsoft Office Project Web Access, que es la versión gratuita de Microsoft Office accesible vía web, la cual contiene los paquetes Word Web App, Excel Web App, PowerPoint Web App, y OneNote Web App, permitiendo a los usuarios acceder, compartir y colaborar sobre los documentos del proyecto. Está diseñado para que pueda ser usado por todos los integrantes del equipo del proyecto, tanto los administradores, como cualquier otra persona que necesite acceso a los datos de Project Server. Office Project Web Access es, en esencia, un conjunto de aplicaciones de Microsoft ASP.NET 2.0 que usan Project Server Interface (PSI).
2. Windows SharePoint Services 3.0 es una aplicación Web de colaboración en grupo y administración de documentos que permite acceder a espacios de trabajo compartidos, almacenes de información y documentos, así como alojar aplicaciones tales como wikis y blogs. Todos los usuarios pueden manipular los controles propietarios llamados "web parts" o interactuar con piezas de contenido, como listas y bibliotecas de documentos. Office Project Web Access está incorporado en Windows SharePoint Services 3.0 para mayor facilidad de uso así como para mejorar la administración, simplificar la personalización y facilitar la integración con otras aplicaciones.

7.9.2. El nivel de aplicación de Office Project Server 2007 incluye:

1. **Project Server Interface:** Project Server Interface (PSI) es la interfaz de programación de aplicaciones (API) de Office Project Server 2007. El modelo de objetos Project Server Interface expone el conjunto de funciones, métodos y funcionalidades de Project Server a todas las aplicaciones externas. Microsoft Office Project Professional 2007, Office Project Web Access, línea de negocio y otras aplicaciones de terceros usan la PSI para tener acceso a los datos de Project Server en las bases de datos. PSI reemplaza Project Data Service (PDS) en Project 2003 pero proporciona paridad funcional para los métodos PDS basados en XML en Project Server 2003, lo que permite trasladar fácilmente las soluciones creadas con PDS

2. **Plataforma de Project Server:** esta plataforma incluye los objetos de negocios de Project Server a los que tienen acceso los usuarios a través del PSI.
3. **Servicios de Project Server** estos servicios son el servicio de informes de servidor, el servicio de eventos del servidor y el servicio de cola de servidor.

7.9.3. El nivel de base de datos de Office Project Server 2007.

Incluye las siguientes bases de datos relacionadas de Project Server:

1. **Borrador:** la base de datos Borrador contiene tablas para guardar los proyectos sin publicar de Office Project Professional 2007. El acceso a los datos de proyectos en la base de datos Borrador no es posible mediante Office Project Web Access.
2. **Publicados:** la base de datos Publicados contiene todos los proyectos publicados. Los proyectos publicados se pueden ver en Office Project Web Access. La base de datos Publicados también contiene tablas que son específicas de Office Project Web Access (partes de horas, vistas, etc.) y tablas de datos globales (códigos de esquema, seguridad y metadatos).
3. **Archivo:** la base de datos Archivo tiene por objeto guardar copias de seguridad y versiones anteriores de los proyectos.
4. **Informes:** la base de datos Informes es el área de almacenamiento provisional para generar informes. Los datos de la base de datos Informes son completos, están optimizados para la generación de informes de solo lectura y están actualizados casi en tiempo real.

Como principales características de Microsoft Project Server 2007 podemos destacar:

Administración de Proyectos:

- Al tener los proyectos almacenados en un servidor permite un acceso ágil y en todo momento de la información necesaria.
- Permite realizar planes de actividades y propuestas de proyectos vía web (a través de Project Web Access), permitiendo visualizar el estado

de la cartera de proyectos, facilitando la toma de decisiones, y acceder a las propuestas realizadas desde cualquier dispositivo con navegador web.

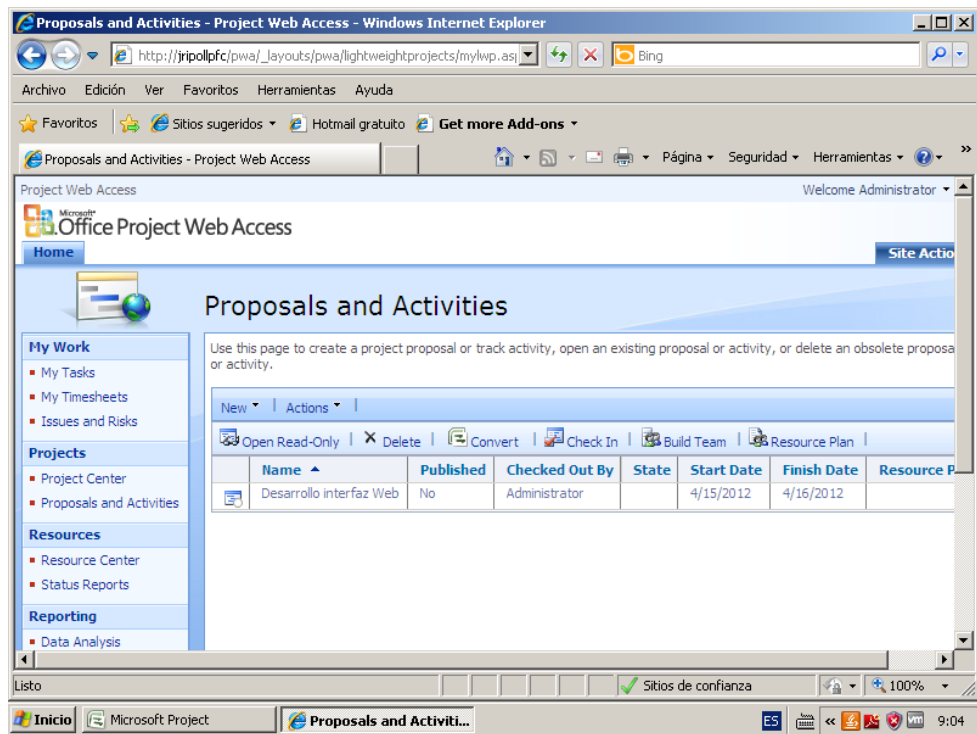


Ilustración 50 Cartera de proyectos activos.

- Área de trabajo del proyecto, que permite identificar objetivos clave con fechas de finalización para los proyectos. También permite asociar compromisos adquiridos con el cliente asociándolos con las tareas del proyecto
- Permite una gestión de proyectos basada en metodologías ágiles, poniendo especial énfasis en la metodología Scrum, existiendo un plugin¹⁴ que dota a Microsoft Project de todas las características de dicha metodología.

¹⁴ Descargar plugin: <http://www.microsoft.com/en-us/download/details.aspx?id=12678>

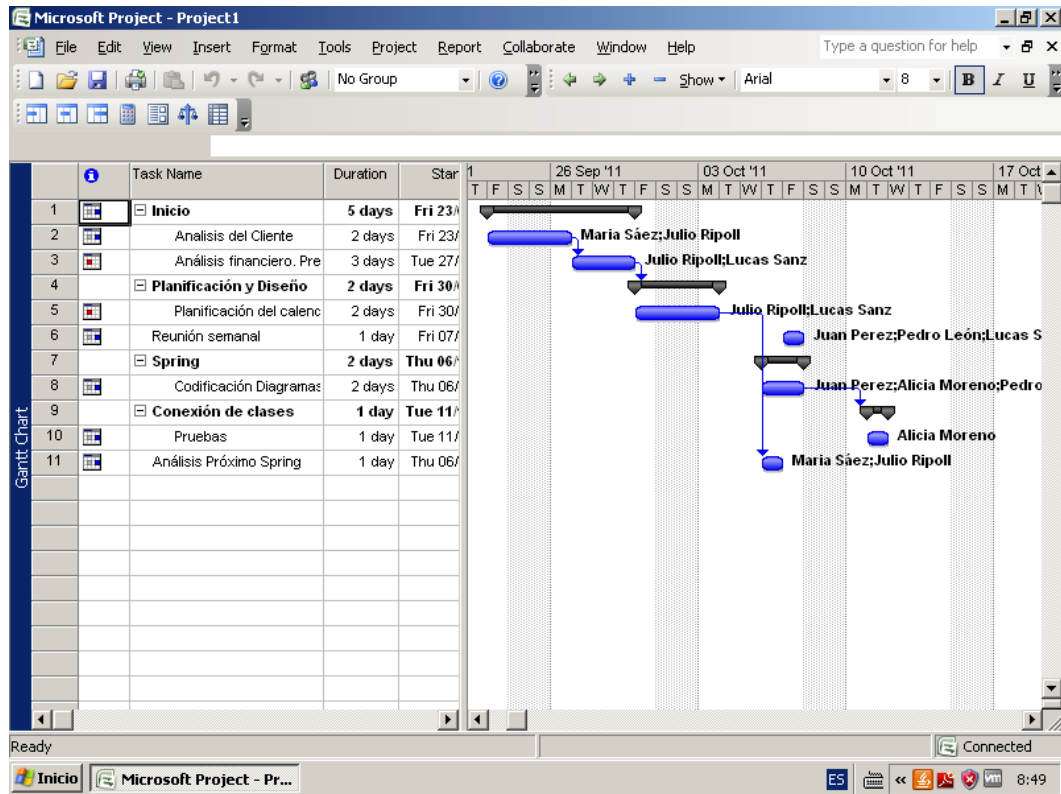


Ilustración 51 Planificación de tareas de un proyecto.

Administración de recursos:

- Permite la definición de recursos, tanto humanos como materiales, de la empresa a través de Project Web Access o PSI.
- Los recursos pueden ser definidos independientemente de la asignación a un proyecto, permitiendo definir las particularidades de cada recurso.
- Permite definir calendarios de la explotación de cada recurso, o tipo de recurso, pudiéndose obtener la disponibilidad del recurso o su sobreasignación.
- Se permite formar equipos de recursos, los cuales pueden ser asignados por completo a un proyecto.

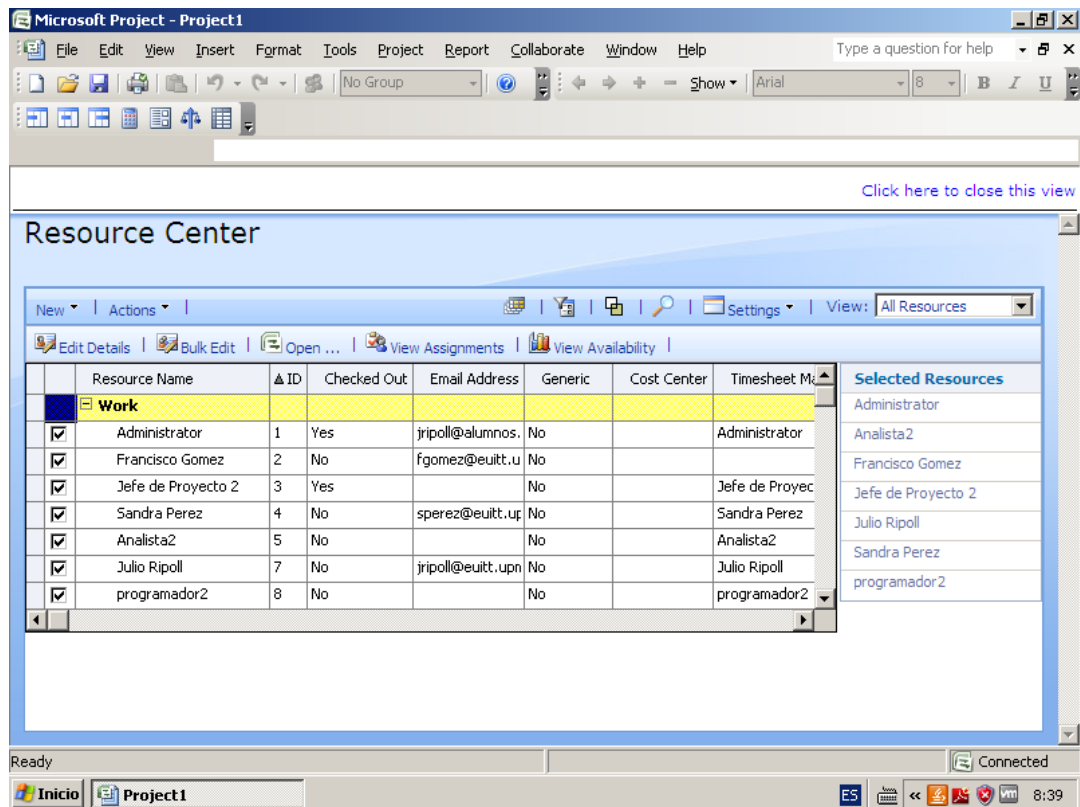


Ilustración 52 Administración de recursos.

Informes:

- Microsoft Project Server 2007 cuenta con una base de datos propia para la realización de informes, la cual es compatible con SQL Server Reporting Services y Servicios de Excel en Microsoft Office SharePoint Server 2007
- Los informes pueden categorizarse en cuatro tipos:
 - De uso de tareas, basados en los datos de tarea con fase temporal, es decir, información sobre tareas, recursos o asignaciones en el transcurso del tiempo (exportables a Microsoft Excel).
 - De uso de recursos, los cuales permiten visualizar diagramas que muestran los costos planeados y reales del proyecto a lo largo del tiempo, el trabajo y la disponibilidad restante de los recursos del proyecto, desglosados por tipo de recursos (trabajo, material y costo), la división del costo de recursos entre los tres tipos de recursos: costo, material y de trabajo, capacidad total, de trabajo y disponibilidad restante de los recursos de trabajo ilustrados a lo

largo del tiempo, o en unidades de trabajo (Microsoft Excel y Visio).

- De uso de la asignación, permiten ver una comparativa en trabajo, material y costo de los dos tipos anteriores, es decir, de tareas junto con los recursos asignados.

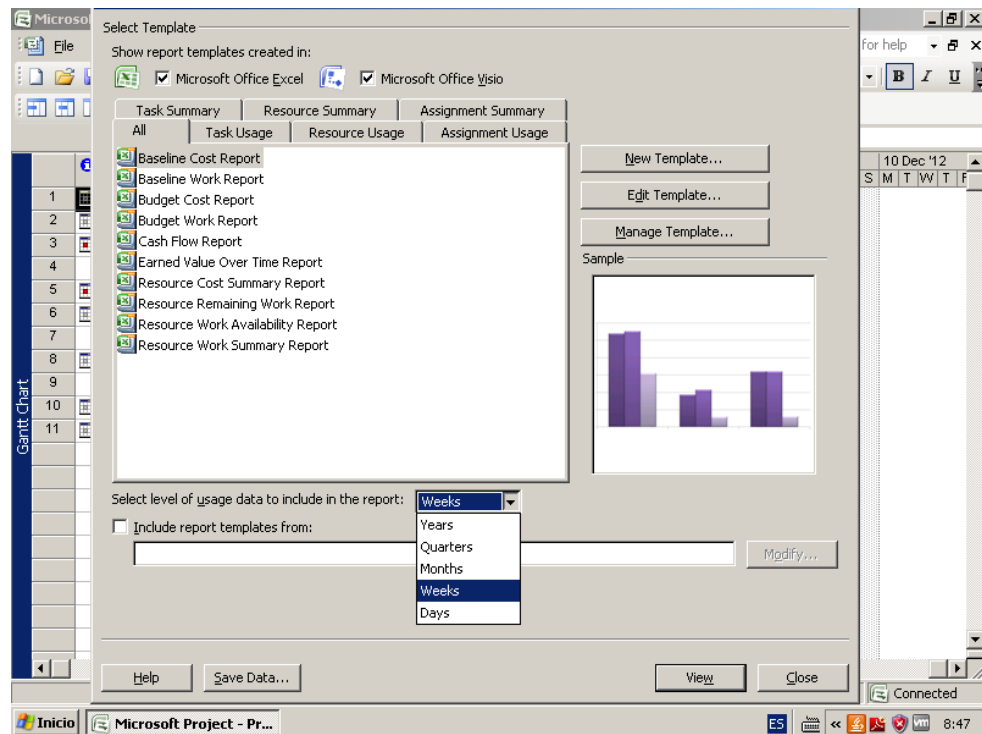


Ilustración 53 Creación de reportes para un proyecto.

Comunicación:

Microsoft Project Server realiza una fuerte apuesta por la comunicación de los miembros integrantes de los equipos de desarrollo y lo hace basándose en Windows SharePoint Services 3.0 y su Area de trabajo, la cual es la ubicación de la información de los proyectos. La plantilla de aplicación proporciona un lugar en el que un grupo puede enumerar, ver tareas y problemas de proyectos, pudiendo crear rápidamente listas compartidas de tareas y distribuirlas, colaborar en la creación de documentos y realizar un seguimiento de los riesgos y problemas clave, manteniéndose constantemente informados de los cambios a través de notificaciones RSS (Really Simple Syndication).

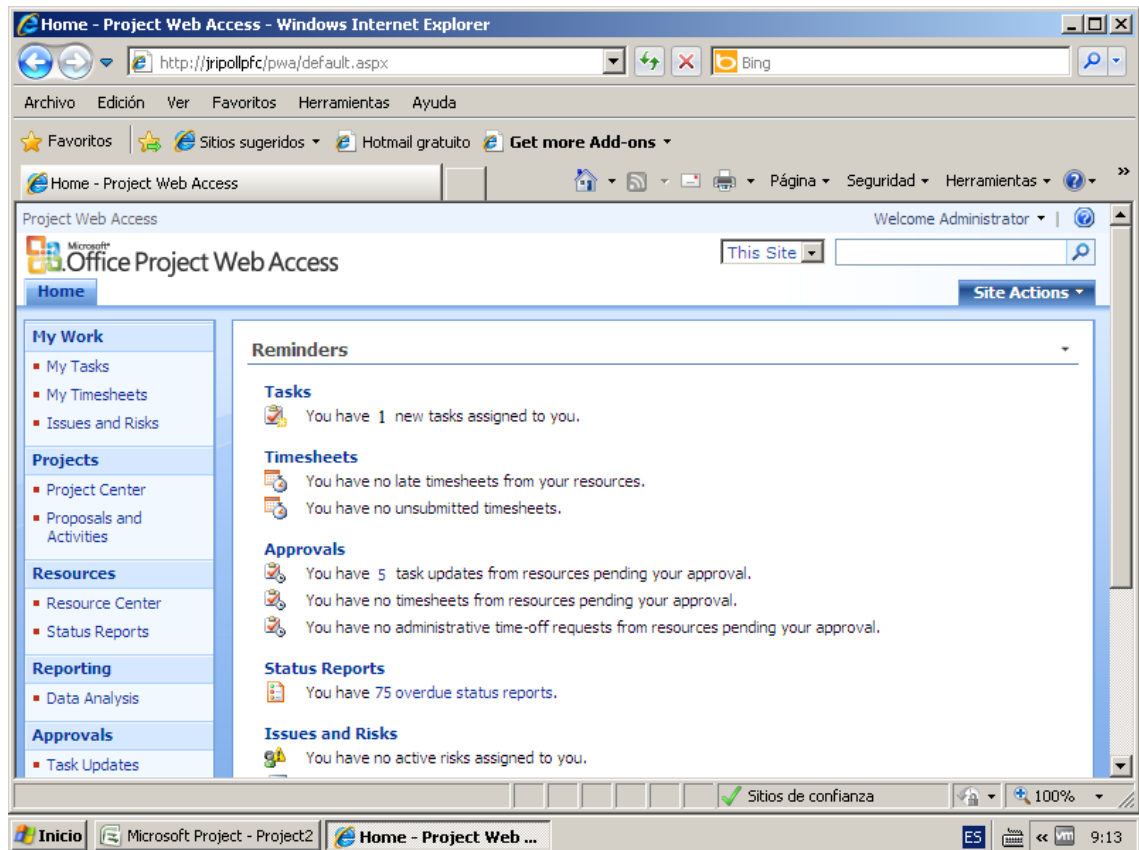


Ilustración 54 Panel de comunicación en Project Web Acces.

Esta aplicación ha sido evaluada con dos máquinas virtuales, una primera, con un sistema operativo Microsoft Server 2003 que contiene todo el paquete de aplicaciones necesarias para hacer funcionar Microsoft Project Server 2007, y una segunda máquina, con sistema operativo Windows XP, en la cual se instaló Microsoft Project en su versión de escritorio, con la cual se sincronizó con Project Server 2007 y se realizaron todas las pruebas de acceso web a la interfaz web de Microsoft Project Server 2007.

Las licencias utilizadas en ambos casos se han obtenido gratuitamente del programa de colaboración entre Microsoft y la Universidad Politécnica de Madrid denominado Microsoft DreamSpark for Academic Institutions.

7.9.4. Instalación Microsoft Project Server 2007:

Para la instalación de Microsoft Project Server partimos de una instalación limpia de Windows 2003 Server Enterprise SP1. Para poder instalar Office Project Server 2007 hay que preparar el sistema, ya que con la configuración inicial no permite la instalación:

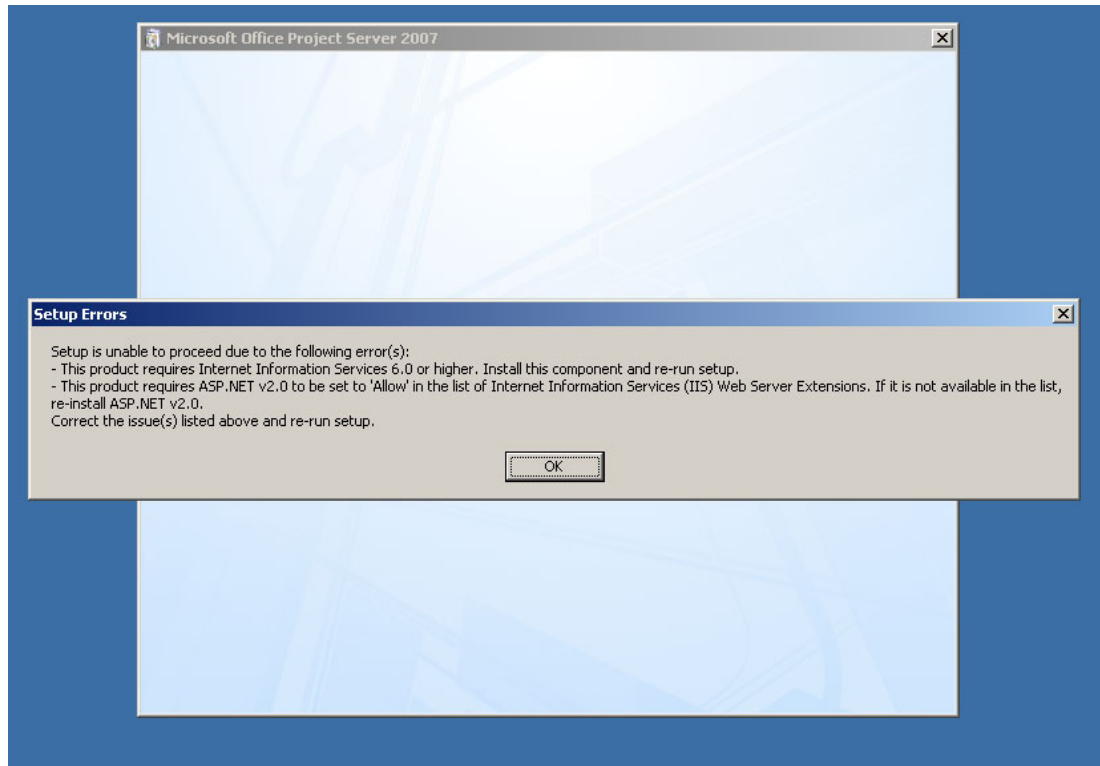


Ilustración 55 Requisitos de instalación.

A continuación se muestra el procedimiento:

1. Lo primero que hay que hacer es actualizar el sistema operativo con el service pack 2 para Windows 2003 server. Los descargamos de la siguiente dirección: <http://technet.microsoft.com/en-us/windowsserver/bb229701.aspx> y lo instalamos.
2. El siguiente paso es instalar Internet Information Services (ISS).
 - a. Para ello hay que hacer clic en Inicio, Panel de control y, a continuación, hay que hacer clic en Agregar o quitar programas.
 - b. En Agregar o quitar programas, hacer clic en Agregar o quitar componentes de Windows.
 - c. En el Asistente para componentes de Windows, en la lista Componentes, seleccione Servidor de aplicaciones.

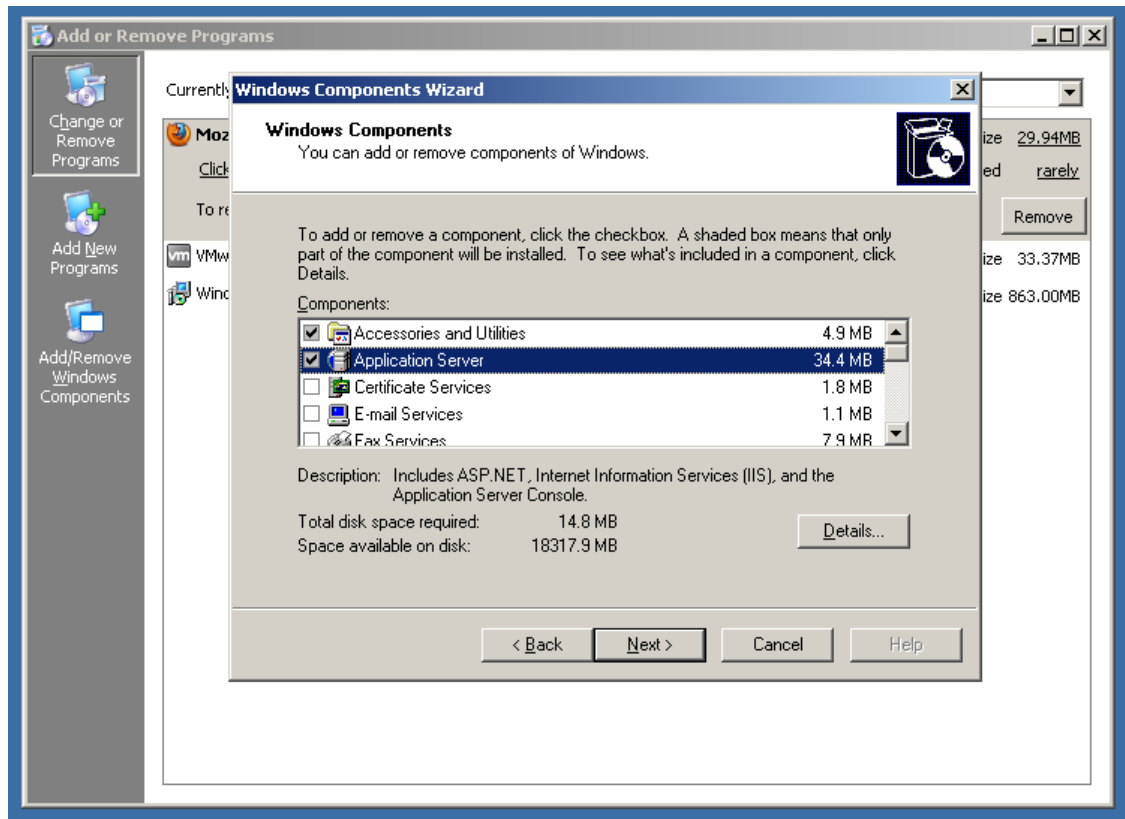


Ilustración 56 Instalación del servidor de aplicaciones.

- d. Hacer clic en Siguiente.
- e. Cuando el asistente complete la instalación, hacer clic en Finalizar.

3. Instalar el Framework 3.5: Para poder instalarlo hay que descargar el archivo de la siguiente dirección:

<http://www.microsoft.com/downloads/en/details.aspx?familyid=333325fd-ae52-4e35-b531-508d977d32a6&displaylang=en> y proceder a su instalación.

Tras esto, debemos permitir la ejecución de ASP .NET en las extensiones del servicio web. Para ello accedemos al ISS Manager:

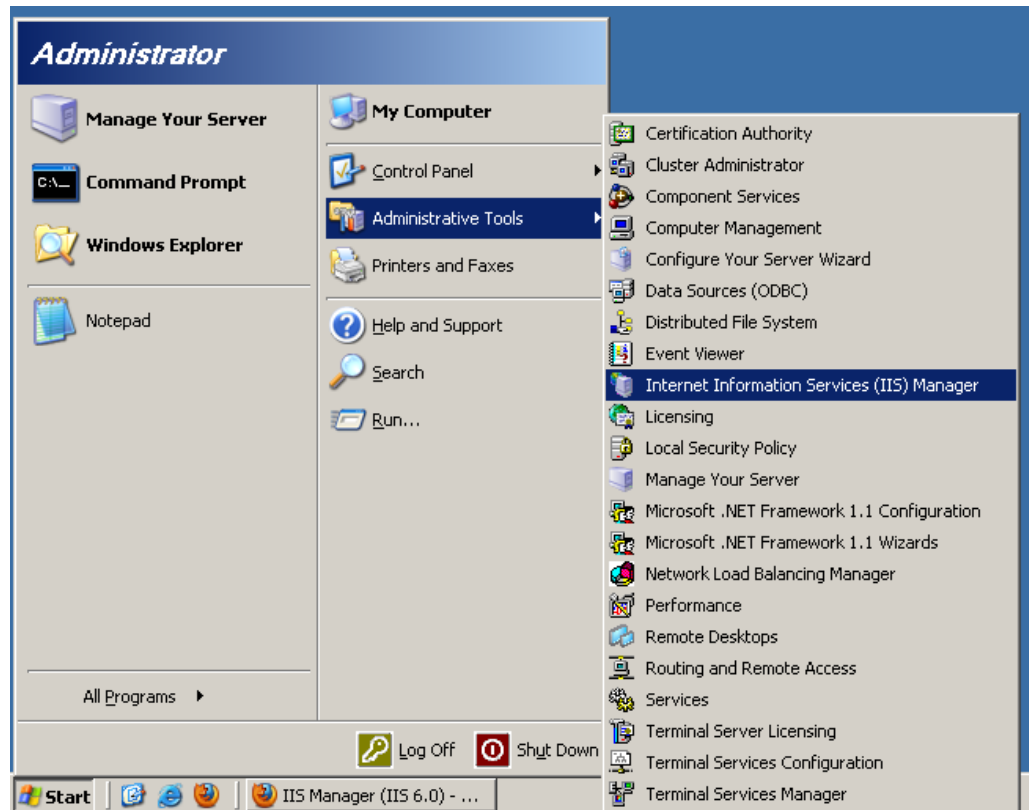


Ilustración 57 Acceso a Internet Information Services Manager.

Tras esto, damos permisos a ASP .NET en Web Service Extensions:

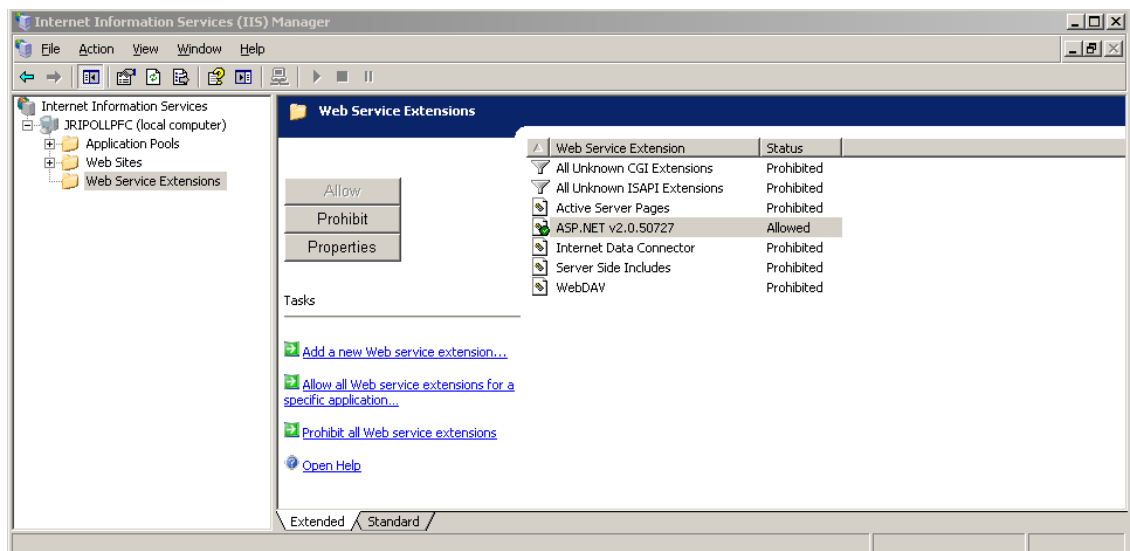


Ilustración 58 Dar permiso a ISS para que ejecute ASP .NET

Finalmente ya tenemos el equipo preparado para comenzar con la instalación de Microsoft Office Sharepoint Server 2007.

Arrancamos el archivo ejecutable y observaremos la siguiente pantalla:

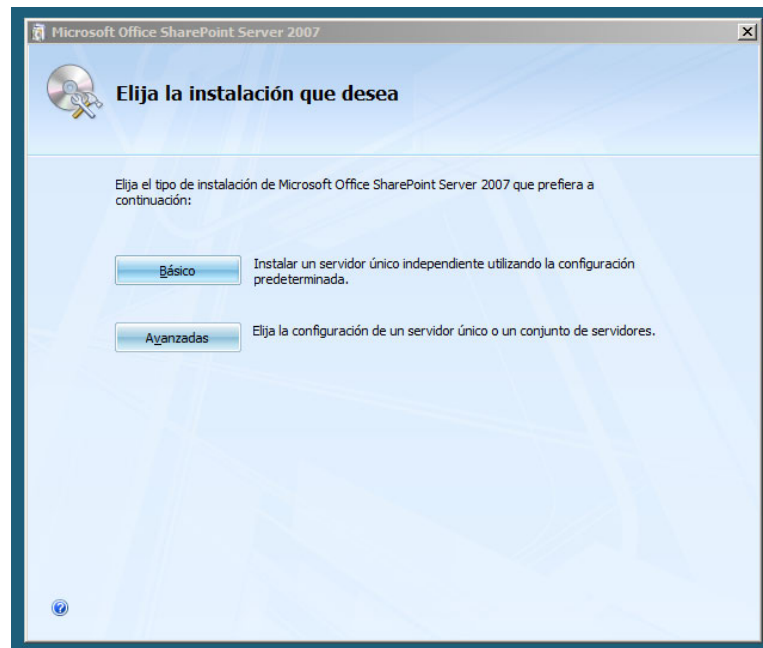


Ilustración 59 Elección del modo de instalación

Elegimos la opción Avanzadas y seleccionamos la opción Independiente, la cual nos permitirá tener un servidor con Office Server 2007, con la base de datos SQL necesaria y Office Sharepoint Server.

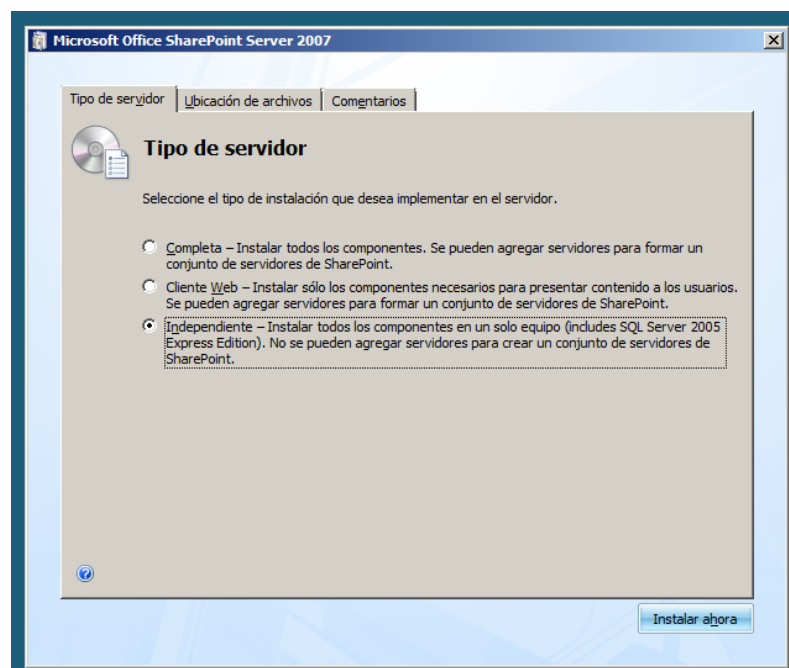


Ilustración 60 Elección del tipo de servidor.

La ubicación de los archivos lo dejamos en los lugares predeterminados. Ahora pulsamos Instalar ahora y comienza la instalación.

Tras la instalación seleccionamos la opción de ejecutar el asistente para el Office SharePoint Server

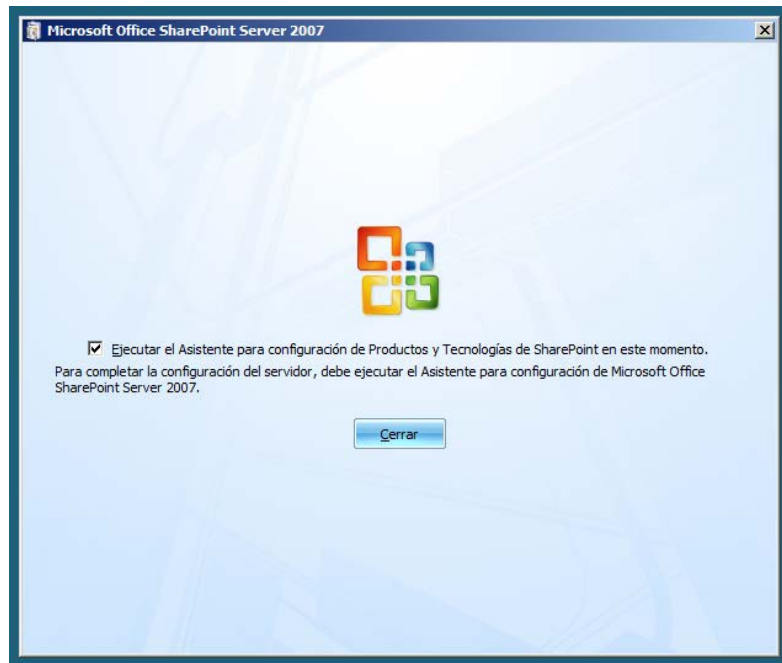


Ilustración 61 Instalador de Microsoft Office SharePoint Server.

El asistente nos instalará todo lo necesario automáticamente.

Tras finalizar el asistente, tenemos el servidor perfectamente configurado para comenzar a utilizar Microsoft Project Server 2007. Para ello basta con acceder a la siguiente dirección en Internet Explorer: <http://nombreDelEquipo/pwa> .

Si se desea configurar alguna opción del servidor web, basta con acceder a la siguiente dirección:

<http://nombreDelEquipo:45585/ssp/admin/layouts/managepwa.aspx>

y configurar los parámetros deseados.

Por último indicar que la autenticación como administrador tanto en Project Server como en la administración del servidor web se realiza con los usuarios y contraseñas definidos localmente en el servidor.

NOTAS: En los clientes es necesario instalar **Project Web Access ActiveX Control** para poder utilizar Office Project Server 2007.

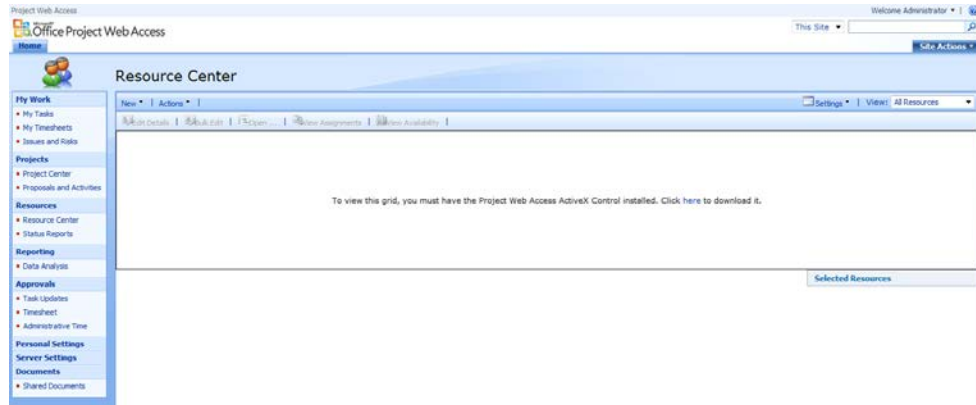


Ilustración 62 Mensaje de advertencia sobre la necesidad de instalar Project Web Access.

Para proceder a la instalación pulsamos sobre el enlace facilitado en el mensaje de advertencia. Posteriormente se abrirá la siguiente advertencia de seguridad:

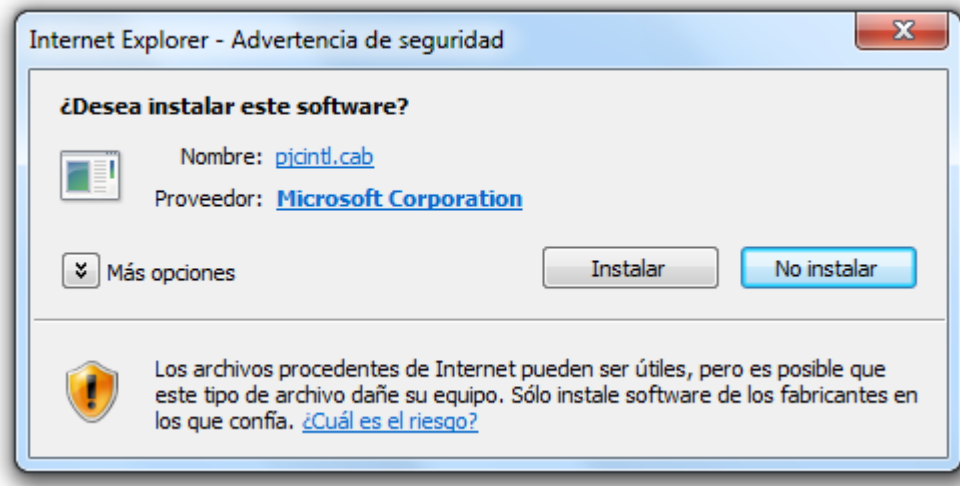


Ilustración 63 Advertencia instalación Project Web Access ActiveX

Pulsamos sobre instalar y se procede a instalar Project Web Access ActiveX Control automáticamente.

8

Bibliografía

- [1] R. A. d. I. Lengua, «Diccionario de la Lengua Española,» [En línea]. Available: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=proyecto. [Último acceso: 2011 Febrero 12].
- [2] M. EYSSAUTIER DE LA MORA, Metodología de la investigación:, 5 ed., CENGAGE LEARNING, 2006, p. 97.
- [3] P. M. I. S. Committee y W. R. Duncan, Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK), Cuarta ed., Pennsylvania: Project Magnament Institute, 2009.
- [4] Peter Nathan; Gerald Everett Jones, PMP Certification For Dummies, Indianapolis: John Wiley & Sons Inc, 2003, p. 63.
- [5] H. Kerzner, Project Management: a systems approach to planning, scheduling and controlling., Hoboken, new Jersey: John Wiley and Sons, Inc, 2009.
- [6] J. P. Lewis, The project manager's desk reference: a comprehensive guide to project planning, scheduling, evaluation, and systems, Berkshire: McGraw-Hill Professional, 2000, p. 185.
- [7] P. S. Committee y W. R. Duncan, A Guide to the Project Management Body of Knowledge (PMBOK Guide), Philadelphia: Project Management Institute, 2008, pp. 27-35.
- [8] A. Aliance, «Agile Manifesto,» Ward Cunningham, Marzo 2001. [En línea]. Available: <http://www.agilemanifesto.org/iso/es/principles.html>. [Último acceso: Febrero 7 2011].
- [9] K. Beck, Extreme programming explained: embrace change, Segunda ed., Boston: Addison-Wesley Professional, 2004, p. 29.
- [10] A. Cockburn, Surviving Object-Oriented Projects, Massachusetts: Addison-Wesley Professiona, 1998.
- [11] A. Cockburn, Agile Software Development, Massachusetts: Addison-Wesley Professional, 2001.
- [12] J. A. Highsmith, Adaptive software development: a collaborative approach to managing complex systems, Nueva York: Dorset House Pub., 2000.
- [13] K. Schwaber y M. Beedle, Agile software development with Scrum, Boston: Pearson Education, Inc., 2008.
- [14] N. Harrison, B. Foote y H. Rohnert, Pattern languages of program design 4. Software Patterns Series, Addison-Wesley, 2000.
- [15] S. R. Palmer y J. M. Felsing, A practical guide to feature-driven development. The Coad series, Upper Saddle River: Prentice Hall PTR, 2002.
- [16] P. Coad, E. Lefebvre y J. De Luca, Java Modeling in Color with UML : Enterprise Components and Processes, California: Prentice Hall PTR, 1999.
- [17] J. Stapleton, DSDM, Dynamic Systems Development Method: The Method in Practice, Addison Wesley, 1997.
- [18] K. Fogel, «Desarrollo de Código Abierto con CVS,» 24 Noviembre 2004. [En línea]. Available: <http://www.linux-cd.com.ar/manuales/cvsbook/index.html>. [Último acceso: 22 Febrero 2011].
- [19] «Repositorio de código fuente del SIDAR,» www.sidar.org, 7 Octubre 1997. [En línea].

- Available: <http://www.sidar.org/recur/desdi/cvs/>. [Último acceso: Febrero 23 2011].
- [20] C. Larman, Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and the Unified Process., Segunda ed., Prentice-Hall, 2002..
 - [21] S. W. Ambler, Agile modeling: effective practices for eXtreme programming and the unified process, New York: John Wiley & Sons, Inc., 2002.
 - [22] S. W. Ambler, «Agile Modeling and the Rational Unified Process (RUP),» 27 Noviembre 2010. [En línea]. Available: <http://www.agilemodeling.com/essays/agileModelingRUP.htm>. [Último acceso: 12 Febrero 2011].
 - [23] A. Hunt y D. Thomas, The Pragmatic Programmer: From Journeyman to Master, Addison-Wesley Professional, 1999.
 - [24] P. Abrahamsson, O. Salo, J. Ronkainen y J. Warsta, Agile Software Development Methods: Review and Analysis., VTT Elekroniikka, 2002.
 - [25] M. Fowler, «The New Methodology,» martinfowler.com, 19 Septiembre 2011. [En línea]. Available: <http://martinfowler.com/articles/newMethodology.html>. [Último acceso: 23 Septiembre 2011].
 - [26] J. W. Reeves, «The (B)Leading Edge,» [En línea]. Available: <http://www.bleading-edge.com/Publications/C++Journal/Cpjour2.htm>. [Último acceso: Marzo 12 2011].
 - [27] E. Miranda, Running the Successful Hi-Tech Project Office, Norwood: Artech House, Inc., 2003.
 - [28] O. o. I. a. T. o. D. o. v. affairs, «<http://www.oit.va.gov/>,» 3 Marzo 2005. [En línea]. Available: http://web.kvif.bgf.hu/upload/subject/doc/20100406123831M_VA_IT_PM_Guide.pdf. [Último acceso: 12 Febrero 2011].
 - [29] J. Becker, M. Kugeler y M. Rosemann, Process Management: A Guide for the Design of Business Processes, Berlin: Springer, 2003.
 - [30] B. Schlagheck, Objektorientierte Referenzmodelle für das Prozeß- und Projektcontrolling : Grundlagen, Konstruktion, Anwendungsmöglichkeiten, Münster: Deutscher Universitäts-Verlag, 2000.
 - [31] J. E. Josef E. Riedl, Projekt - Controlling de Forschung und Entwicklung, Berlín: Springer-Verlag, 1990, p. 99.
 - [32] S. Bruch Lawa, Projektmanagement, Frankfurt: FAZ Verlagsbereich Wirtschaftsbücher, 1995.